Reliable and Error Tolerant Handwritten Numeral Classifier by Rejecting Undesirable Samples From the Training Set

David Yu

A Thesis

In the Department

of

Computer Science and Software Engineering

Presented In Partial Fulfillment Of The Requirements

For The Degree of Master of Applied Sciences at

Concordia University

Montreal, Quebec, Canada

August 2012

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: David Yu

Entitled: Reliable and Error Tolerant Handwritten Numeral Classifier by Rejecting Undesirable Samples From the Training Set

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Sciences

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

____________ Dr. V. Haarslev _______	Chair
____________ Dr. T. D. Bui _________	Examiner
____________ Dr. A. Krzyzak _______	Examiner
____________ Dr. C. Y. Suen ________	Supervisor

Approved by ____
Chair of Department or Graduate Program Director

____
Dean of Faculty

Date ________________ September 25, 2012 _______________

ABSTRACT

Reliable and Error Tolerant Handwritten Numeral Classifier by Rejecting Undesirable Samples From the Training Set

David Yu

Unconstrained handwritten numeral recognition has many applications which include: reading handwritten bank checks, extracting numbers from tax forms and sorting ZIP codes on letter mail. However, these automated systems still make mistakes and any resulting corrections can be expensive. The main focus of most research in this field is to improve recognition accuracy, whereas the reliability rate has been neglected. Our goal is to build a system that is 100% reliable (no errors) while maintaining a high recognition rate.

A very common strategy to achieve better reliability is to implement a rejection mechanism, which processes the predicted results from a classification system. A novel rejection approach that increases the reliability and accuracy of current classifications systems is proposed in this thesis.

Our thesis compares the effect on both recognition and reliability rates using the training set rejection system, post-testing rejection system and combining both models on a classifier. A two-stage rejection system is proposed to improve the reliability of a classifier. The first stage of the rejection system purifies the training set of undesirable samples. The second stage of the rejection system removes

results that do not have a strong correlation with their respective class. Our experiments study the effect on both recognition rate and substitution rate by employing a structural feature and a statistical feature. The study is conducted over the popular MNIST database for easier comparison with other methods. We are using a support vector machine based classifier, as it has achieved one of the best recognition systems to date.

Lastly, a category system is created to identify the types of samples removed from the training set. The samples are categorized into six major groups: good, very slanted, thick stroke, poor, unrecognizable, and confusing pairs. The first three categories are desirable samples, and the last three are undesirable samples from the training set. The performance of the classifier shows improvement of up to 0.03% when samples belonging to one or more undesirable groups are removed.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor, Dr. Ching Yee Suen, for his continuous guidance and invaluable support. His encouragements and helpful advice made this thesis possible.

I would like to thank my friend and colleague, Muna Khayyat, for her valuable suggestions and kind encouragement. She was a source of great inspiration, offering constructive suggestions for my thesis.

I wish to thank Dr. Chun Lei He for her help and valuable time. I appreciate your detailed explanations and support.

I want to thank my friends and colleagues at the Centre for Pattern Recognition and Machine Intelligence: Khoa Luu, Malik Waqa Sagheer, Bing Zhang, Graciela Lecireth Meza Lovon, Mohammad Mehdi Haji, Xiaoxiao Niu, Adnan Abueid, Fariba Haghbin, for creating an enjoyable research environment. I am grateful to our research manager, Mr. Nicola Nobile, for his technical guidance and our office assistant, Marleah Blom, for her cheerfulness and administrative assistance.

I am indebted to my AMD colleagues: Omesh Mutukuda, David Egas, and Jenny Liu who proofread this thesis, including this very sentence.

Lastly, I wish to thank my fiancé, Jinzi Huang, my parents and future in-laws for their love, support and understanding. I dedicate this thesis to you.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

This chapter introduces the research topic, motivation, challenge, previous work, proposed solution, and the thesis outline. Section 1.1 describes the current thesis research topic. Section 1.2 explains the motivation behind this research topic. Section 1.3 presents the current and future challenges encountered in this research area. Section 1.4 provides a brief description of work previously completed in this field. Section 1.5 gives an overall view of the proposed solution.

1.1. Research Topic

Pattern recognition encompasses a vast subset of smaller fields including speech recognition, object recognition, character recognition, etc. Handwriting recognition is one of the most developed and explored fields of all. The purpose is to allow a machine to recognize human's handwriting correctly. There are two kinds of handwriting recognition: online and offline. Online handwriting recognition can achieve a higher recognition, as it contains additional data not available in offline handwriting.

The focus of this thesis is on offline unconstrained handwritten numeral recognition. Unconstrained handwritten numeral recognition is an extensively investigated topic and an increasingly challenging problem to perfect.

1.2. Motivation

Handwritten numeral recognition is widely used for various tedious tasks still currently performed by humans. Some of its most common applications include reading handwritten bank checks, extracting numbers from tax forms and sorting ZIP codes on letter mail. Although some are slowly adopting technology to execute these jobs, others are still reluctant to use an automated system. For the reason that computers often commit errors which humans would easily recognize. The challenges of this problem and the methods to address these issues are described in the subsequent sections.

Classifiers use the training set to learn recognizing recurring patterns on an unknown testing set. Consequently, a classifier's performance ultimately depends on the quality of the training set. The topic of optimizing a training set includes preprocessing works such as binarization [1], denoising, slant correction [2], normalization [3], and thinning [4]. Data synthesis, on the other hand, expands the training set using different methods such as affine transformations, elasticity transformations, and virtual support vectors [5]. It is interesting to study the extent to which the performance of a classifier is affected by altering the training set in unusual ways.

1.3. Challenge

As we know, errors are costly in an automated system. If the classifier is not confident with a result, human intervention is required and the task needs to be repeated during manual verification. Errors can also occur and are noticed only at a later stage where corrections can be expensive. This process is very time consuming (to go through all the readings again) and as a result very costly. Ideally, we want a system that can recognize every sample correctly but such a system is impossible to build because even humans can make mistakes. However, it is possible to have a next to perfect automated system that can recognize "readable" samples and reject "unreadable" samples. In this next to perfect world, the machine would make no mistakes and only the rejected samples will need to be reviewed. This would minimize the cost of human intervention in the automated recognition system. Ultimately, we want to maximize the number of recognized samples and minimize the rejected sample set, while committing no errors. Therefore, reliability by error tolerance and prevention is the primary focus of the thesis.

Currently, the system with the highest recognition rate still makes errors if no rejection is performed. However, when rejection is added, the recognition rate would be guaranteed to drop as correctly recognized samples are also rejected. This is the greatest challenge faced in unconstrained handwritten recognition and the topic addressed by this thesis.

1.4. Previous Work

In this section, recent studies of handwritten numeral recognition rate are described. Many types of classifiers have been investigated by researchers to simulate human recognition of characters, such as modified quadratic discriminant function (MQDF), learning vector quantization (LVQ), k-nearest neighbors (KNN), multi-layer perceptron (MLP), radial basis function network (RBF), etc. A comprehensive study and their respective performance on the extended NIST database can be found in the paper by Liu et al [6].

Lately, many improved machine-learning classifiers are extensively used in the pattern recognition field. The two most prominent ones are obviously the convolutional neural network (CNN) and the support vector machines (SVM). Both have attained the highest recognition rate to date. Ciresan et al. have achieved a recognition rate of 99.77% using a CNN classifier and Dong et al. have managed to achieve a recognition rate of 99.44% using HeroSVM, a fast SVM classifier [7] [8]. Readers interested in CNN can refer to a paper published by LeCun, which provides all the details of its inner workings [9]. A general overview of SVM is provided in section 2.4 of this thesis as it is the classifier of choice for this thesis.

It is fascinating that affine and elastic transformation can improve the recognition accuracy to a new level. Simard et al reported reaching a recognition rate as high as 99.60% on MNIST using a CNN classifier [10]. However, there are still improvements to be made for a system to be both reliable and accurate. He and Suen with their hybrid multiple classifier system managed to achieve a reliability rate of 99.93% with a recognition rate of 95.54% [11]. Suen and Tan analyzed the errors of handwritten digits made by a multitude of classifiers [12]. It was concluded that a human being could recognize more than half of the errors easily.

As reviewed in the literature, the general work or principle of rejection is done during the testing stage of a classification system or post-testing stage to avoid making a mistake at that point. Traditionally, the rejection system is appended to a classification system. Based on a set of rejection criteria, a sample is rejected if deemed too risky. Classically known methods include: testing the confidence level of the current recognized class against a threshold, comparing the confidence level of the current recognized class with other possible class and having a committee of classifiers to confirm a sample's identity using majority vote, etc. In a novel approach, our solution will prevent errors from reaching the classifier and provide error tolerance as early as the training stage of the classifier.

1.5. Proposed Solution

Rejection has proven to be useful in removing mistakenly recognized images and have always been applied as a post classification process. However, the classification process is ultimately based on the image set previously used to train the classifier. The training stage occurs when the classifier learns how to categorize each image to the correct class. This concept is analogous to a parent teaching a baby how to recognize objects early in life. Like a baby, the classifier begins as a blank slate and we can shape them any way we want. We can teach the baby in a language of our choice and similarly "teach" a classifier to recognize numerals, letters, or Chinese characters. This is a very critical step for both the baby and the classifier. If we make a mistake as the parent/researcher, this mistake is repeated by the baby/classifier in the future. For instance, if we teach a baby that mom is actually "ga" (a made up word), the baby will refer to mom by "ga" on the next occasion. Similarly, if we train the classifier to recognize a numeral "5" as a"3"; the next time it encounters a numeral "5", it will label it as a "3". This reinforced the importance of the training data set used by any classification system. Therefore, it is imperative that the training samples are free of errors, malformed and unrecognizable images.

Based on the above, it is interesting to study the effect of altering the training set on a classifier. It is expected that the recognition rate of the classifier might be lowered as we remove samples from the training set. This is because the classifier tested has a smaller knowledge base for recognition. From our knowledge, this is the first attempt at investigating the error tolerance and detection capability of handwritten numeral classifiers.

Our approach is very simple, we would like to study the effect of rejecting samples from the training set on different combinations of features and classifiers. A change in the training set can vary the effectiveness of different types of features and in turn impact the result of the recognition system. The feature set is identified as an important contributor to any classifier's recognition rate. Features are extracted from images to simplify and facilitate the recognition process for a classifier. The quantity and quality of the features set selected have a direct impact on a classifier's performance. Thus, it is necessary to test this effect over a wide range of features preferably at least one from each of the following type: statistical and structural.

For statistical feature, we chose to use the projection feature. This feature is simple and efficient for numeral recognition. This feature is based on statistics extracted from the image. The projection feature is one of the best statistical features available as it offers a good accuracy rate for a small number of extracted features.

The gradient feature was selected as the structural feature mainly of its excellent performance for numeral recognition and it also creates a long feature vector. The gradient feature has been widely used within a large range of classifiers and has proven to yield one of the highest accuracy rates to date. This is particularly true when fed to a support vector machine based classifier as demonstrated by He et al [19]. The gradient feature excels at retaining the position, direction, and strength characteristics of the image that is being extracted.

Therefore, different classification methods may react in different ways to this variation from both the training data and feature sets. The classifier used for this experiment is support vector machines (SVM) based.

The extracted features should follow the law of statistics and fit in a normal distribution. The system will identify and reject the outliers, which are samples that have features lying outside of the top and bottom 5% limits of the normal distribution. This percentage comes from the applied practice in statistical hypothesis testing where the typical confidence interval is defined at the 95% confidence level. The limiting percentage will be investigated and varied to test the efficiency of this rejection mechanism. It is suspected that this procedure will lower the accuracy of the system but the main purpose is to eliminate any errors at the cost of recognition rate. In real-life application, errors are more costly to correct and we prefer to reject the sample instead of making a mistake.

The system will first be tested on the MNIST database to obtain some preliminary results using the projection feature. A more detailed analysis will be done while using the gradient feature, as it can yield the best recognition results up to date. The CNN is not tested in this thesis because it is not suitable and naturally compatible with our pre-training rejection mechanism.

1.6. Thesis Outline

This thesis is structured in the following manner:

Chapter 2 offers a brief theoretical background of the database, the features, and the classifier used to conduct our research. Section 2.1 introduces the reader to the MNIST database. Section 2.2 follows with a discussion about image pre-preprocessing techniques used in this thesis such as noise removal, slant correction, normalization, skeletonization, and binarization. Section 2.3 describes the feature selection process and both statistical and structural features studied. Section 2.4 finishes by providing a short overview of support vector machines (SVM) and its performance.

Chapter 3 discusses the results of rejection in the training set for the projection feature using a support vector machine classifier. Section 3.1 describes the architecture of the overall system and the algorithm of our rejection system. Section 3.2 presents the results obtained for rejection in the training set only. Section 3.3 and 3.4 provide an analysis and evaluation of the results for rejection in the training set. Section 3.5 and 3.6 presents the results and analysis for rejection in the testing set only. Section 3.7 and 3.8 describes the results and analysis for rejection in both testing and training sets. Section 3.9 concludes with an evaluation of rejecting in both testing and training sets.

Chapter 4 shows the results of rejection in both the training for the gradient feature with an SVM classifier. Section 4.1 depicts the design of the general classification system. Section 4.2 provides the results obtained along with an analysis. An introduction to the six categories of rejected samples is described. Statistics about the categorized samples are collected. Several hypotheses on the results are formulated and additional results are analyzed to confirm our theory. Section 4.3 concludes the chapter with an evaluation of its performance against the original training set and preceding sections.

Chapter 5 concludes the thesis. The contributions of this thesis are outlined. A final evaluation of the findings using our proposed methodology and its values are discussed. Lastly, future research directions are presented.

Chapter 2: Theory and Background

In this chapter, we present the database that was used for this research, the preprocessing techniques, the feature selection, the projection feature, the gradient feature, and the SVM classifier, which constitute the error preventing and tolerant classification system.

2.1. Database

The MNIST database is a subset of the larger NIST database of unconstrained handwritten digits [13]. It is the most widely used database by researchers in the offline handwritten numeral recognition field to compare their systems performance. Thus, this database is also used to benchmark the performance of the solution proposed in this thesis. MNIST is composed of 60,000 training samples used to train the classifier and another 10,000 testing samples used to verify the classifier's performance. Each sample is a 20 x 20 pixels image that has been centered and size normalized to 28 x 28 pixels. Some samples of images from the MNIST database are shown in the following figure.

Figure 1: Sample images with their labels from the MNIST database.

2.2. Image Pre-processing

It is best to apply pre-processing to images as it usually yields better results than using raw database images for both training and testing purposes. This step is always done before feature extraction and makes the features extracted from the images more consistent and relevant. The goal of pre-processing an image is to reduce the variations in images of the same class and facilitate the following feature extraction process. There are many image pre-processing techniques, such as noise removal, slant correction, normalization, skeletonization, and binarization that are commonly used in pattern recognition. All above listed pre-processing techniques are used in the current thesis. A brief description of each technique is described next.

Noise removal: consist of removing noise, such as salt and pepper from images. This process is done using a mean filter that removes small and unwanted details about an image, also known as blurring. For our purposes, a mean filter of 3x3 is being used to blur out the noise from our database of images.

Slant correction: is used to straighten digits that have been written with a slant. The first step of correcting a slant is to estimate its angle. After estimating the slant angle, a horizontal shear transformation is applied to the image in order to shift them left or right depending if the slant angle has a positive or negative value.

Normalization: consist of making all images homogeneous in size. The images are either enlarged or reduced using linear normalization. The following pair of equations are used: $x' = ax$ and $y' = by$, where a and b are the scaling factors calculated based on the ratio of the target normalization size over the original size in the horizontal and vertical axis respectively. Missing pixels are filled by linear interpolation of adjacent pixels.

Skeletonization: different writing instruments produce numerals of different stroke width. The various stroke thicknesses lead to within class image variations that must be eliminated. Skeletonization or thinning is a pre-processing technique that is used commonly in numeral and character recognition. Thinning is used to reduce the thickness of the image by stripping the outer and inner contour of the image until it is not reducible anymore. This is when we have the basic skeleton that is still recognizable.

Binarization: consist of changing all pixels of an image to either black or white. For example, a grey scale image can contain values from 0 to 255. Each pixel will contain either 1 or 0 after binarization, which reduces its dimensionality. The challenge of binarization is to find a threshold for separating black and white values. Some use a single global value picked between the maximum and minimum value after many trial and error experiments. A down side to this method is that it will not work for images that are either very light or very dark. In these cases, the images would be indistinguishable from the background. A better method is to use Otsu's method for thresholding, which selects a different value for different image [14]. This adaptive method works best for most cases including very light and very dark images, as it can still differentiate the foreground image from the background.

2.3. Feature selection

The performance of a classifier depends on a multitude of factors. The feature set is identified as a top contributor among these factors to any classifier's recognition rate. Features are extracted from images to simplify and facilitate the recognition process for a classifier. This effectively reduces the dimensionality of the problem by filtering unnecessary information while emphasizing the relevant data. During a classifier's training and testing phases, the features are extracted and processed. The quantity and quality of the features set selected have a direct impact on a classifier's performance. An ideal feature would accentuate the intra-class common traits while separating inter-classes distinct traits without losing useful information from the original data. When analyzing the error prevention and error tolerance of the system proposed, two distinct sets of features are used. The following two sub-sections will describe the projection feature set and the gradient feature set used for this experiment.

2.3.1. Projection Feature

Projection is a very efficient feature that is part of the statistical feature set. The ultimate statistical features should ideally have a disjoint set of values for each class to be identified. Since this is not a feasible case in the real world, a good feature set has to minimize the overlapping values for different classes to minimize confusion for the classifier. The projection feature is one of the best statistical features available as it offers a good accuracy rate for a small number of extracted features.

2.3.2. Gradient Feature

The implementation of the gradient feature in this thesis follows the procedure prescribed in the paper published by Shi et al[16].

The gradient feature can be obtained from a gray-scaled image. The binary image first needs to be converted to a gray-scaled image using a 3x3 mean filter. For the MNIST database, it would be counter productive to binarize an image and then convert it back to grayscale. Therefore, the above steps are omitted in our experiment. The image is then normalized. This normalized image of size 32x32 is then boxed in a zero padded 2 pixels wide frame and centered as a 36x36 image. The gradient's strength and direction vectors are obtained by applying the following formulae on each pixel g(m,n):

$$\text{Direction: } \theta(m,n) = \tan^{-1}\frac{\Delta v}{\Delta u}$$

$$\text{Strength: } s(m,n) = \sqrt{\Delta u^2 + \Delta v^2}$$

where

$$\Delta u = g(m,n) - g(m+1,n+1)$$

$$\Delta v = g(m,n+1) - g(m+1,n)$$

and where θ (m, n) and s (m, n) denote respectively the gradient's direction and gradient's strength of pixel g (m, n). The image is then divided into 81 blocks of 4 x 4 pixels each delimited by nine horizontal and nine vertical lines. The strength and direction of the gradient are combined to compute a single feature vector. The direction of the gradient is quantized to 32 levels over the 2π range with π /16 intervals. For each of the 81 blocks, the strength of the gradient in each of the 32 directions is calculated. A large feature vector containing directions and directional magnitudes of size 81 x 32 is obtained. However, such a large feature vector is not desirable, as it will take a large amount of time to train and test. It is recommended to down sample the feature vector produced earlier. The down sampling process is done in two steps. First, the original 32 directions are reduced to 16 directions, each with π /16 intervals by combining pairs of adjacent directions. This effectively reduces the feature vector to a size of 81 x 16. After, a Gaussian filter of size 5x5 is applied to the 81 blocks (9 x 9) in order to reduce the vector size to 25 x 16 or 400 features. A transformation $y = x^{0.4}$, is then applied to the feature vector to make it more Gaussian as suggested by Shi [16]. Finally, the feature vector is normalized to values between zero and one. In the next figure, we show an example of a normalized grayscale image in (a), its gradient strength in (b), and gradient direction in (c).

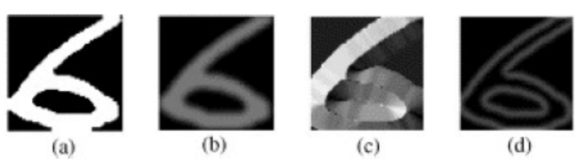


Figure 2: a) numeral binary image, b) gray scale image, c) direction of gradient, d) strength of gradient [16].

2.4. SVM

This section describes some basic concepts of the support vector machines (SVM) and how it functions as a classifier.

2.4.1. Description

The most basic SVM consists of supervised learning methods, which are used to separate two classes of objects apart statistically. SVM, during its training process delimits members of different classes by drawing hyperplanes in two or more dimensions. During this process, the SVM constantly readjust its hyperplanes in order to maximize the dimensional distance between each classes and the hyperplanes, also commonly known as function margin. A SVM separating classes with a maximal margin would yield better recognition rate than one that has a smaller margin. The following figure depicts this process.

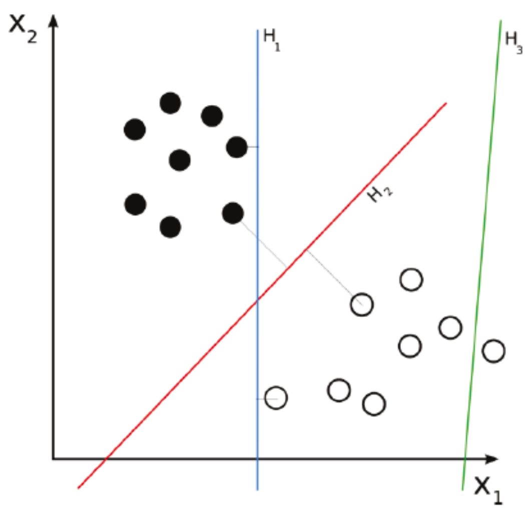


Figure 3: Separation of classes using different hyperplanes or lines in 2 dimensions [17].

The figure above shows three hyperplanes H1, H2, and H3. H3 in this case is a bad choice, as it does not separate the two classes. H1 is a better choice as it is separating the two classes for the features shown, but the inter class distance is not maximized. H2 is the best hyperplane possible for this situation. If we compare the distance between each class sample's that is closest to the hyperplane for both H2 and H3, we can observe that it is over five times larger. It means any object that falls in between the H2 and H3 hyperplanes will have a high probability of being misclassified if using H2 as hyperplane. A higher dimensional space is used to compute the hyperplane in the following figure.

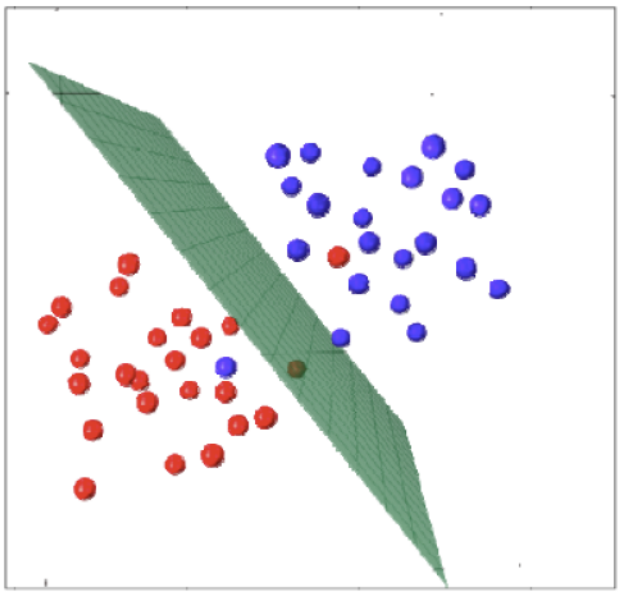


Figure 4: Separation of classes using a hyperplane in 3 dimensions [15].

From the training set D with n data points, the following equation is formulated:

$$D = \{(x_i, y_i) | x_i \in \mathbb{R}^p, y_i \in (-1, 1)\}_{i=1}^{n}$$

indicating the class to which the point *xi* belongs based on value *yi*. When *yi* = -1, it is a member of class 1 and when *yi* = 1, it is a member of class 2. Each *xi* is a *p*-dimensional real vector [17]. To find the maximum-margin hyperplane that separates the points in one class from the points in another class, we need to solve the following optimization problem:

$$y_i\,(w.x_i - b) \geq 1$$

where w has to be as small as possible.

2.4.2. LIBSVM

LIBSVM is a library for Support Vector Machines developed by Chang and Lin [18]. It supports multiclass classification and is available in a variety of programming languages such as C++, Java, and MATLAB, which are all maintained by the authors. LIBSVM is straightforward and provides easy to use module interfaces that can be smoothly integrated into a system. Some features of LIBSVM include [18]:

- Different SVM formulations
- Efficient multi-class classification
- Cross validation for model selection
- Probability estimates
- Various kernels (including precomputed kernel matrix)
- Weighted SVM for unbalanced data

LIBSVM is used in our experiments to predict class labels and to provide individual class probabilities. The core kernel function of the SVM we have selected to use for this thesis is the radial basis function (RBF).

2.4.3. Performance

Support vector machines based classifiers have shown high recognition rates in various applications for pattern recognition. SVM is widely used in text classification, face recognition and handwritten digit recognition. The following table shows a list of the highest recognition accuracies achieved using only a support vector machines based classifier on handwritten numeral databases.

Table 1: Summary of performance achieved using SVM based classifiers on handwritten numerals databases [20].

Author	Recognition Rate (%)	Error Rate (%)	Rejection Rate(%)	Database
Dong et al	98.70	1.30	0.00	CENPARMI
Oliverira et al	99.20	0.80	0.00	NIST SD 19
Teow et al	99.41	0.59	0.00	MNIST
Liu et al	99.58	0.42	0.00	MNIST
DeCoste et al	99.58	0.42	0.00	MNIST
Dong et al	99.62	0.38	0.00	MNIST

As observed above, SVM based classifiers are very efficient and hence are a natural choice to test our proposed solution.

Chapter 3: Rejection in the Training Set and Testing Set on Statistical Features

In this chapter, a rejection system in both training and testing sets for statistical features using an SVM based classification system is proposed for handwritten numerals recognition.

A classifier's accuracy depends highly on its training set. Thus, research on improving the invariability of the training set such as slant correction, and normalization are often used to preprocess both the training and testing set. Another method, such as training set expansion by means of data morphing uses affine and elastic transformation to achieve higher recognition rate (99.6%) [10].

In this thesis, we suggest that detecting and rejecting offending samples directly from the training set can achieve a more reliable recognition system. We will investigate the effect of combining our novel training set rejection system with a similar rejection system on the testing set. Since detecting and rejecting offending samples directly from the training set may achieve a more reliable recognition system, we suggest applying the same process to the testing. This should accumulate the benefits from both rejection systems and create a very reliable classification system with error prevention and error tolerance attributes.

3.1. Methodology

The training set rejecting classification system uses features statistics to identify outliers in the training set. This method is applied after the training set has gone through the image pre-preprocessing and features extraction procedures. The following picture provides a better view of the overall system that is being investigated first in this section.

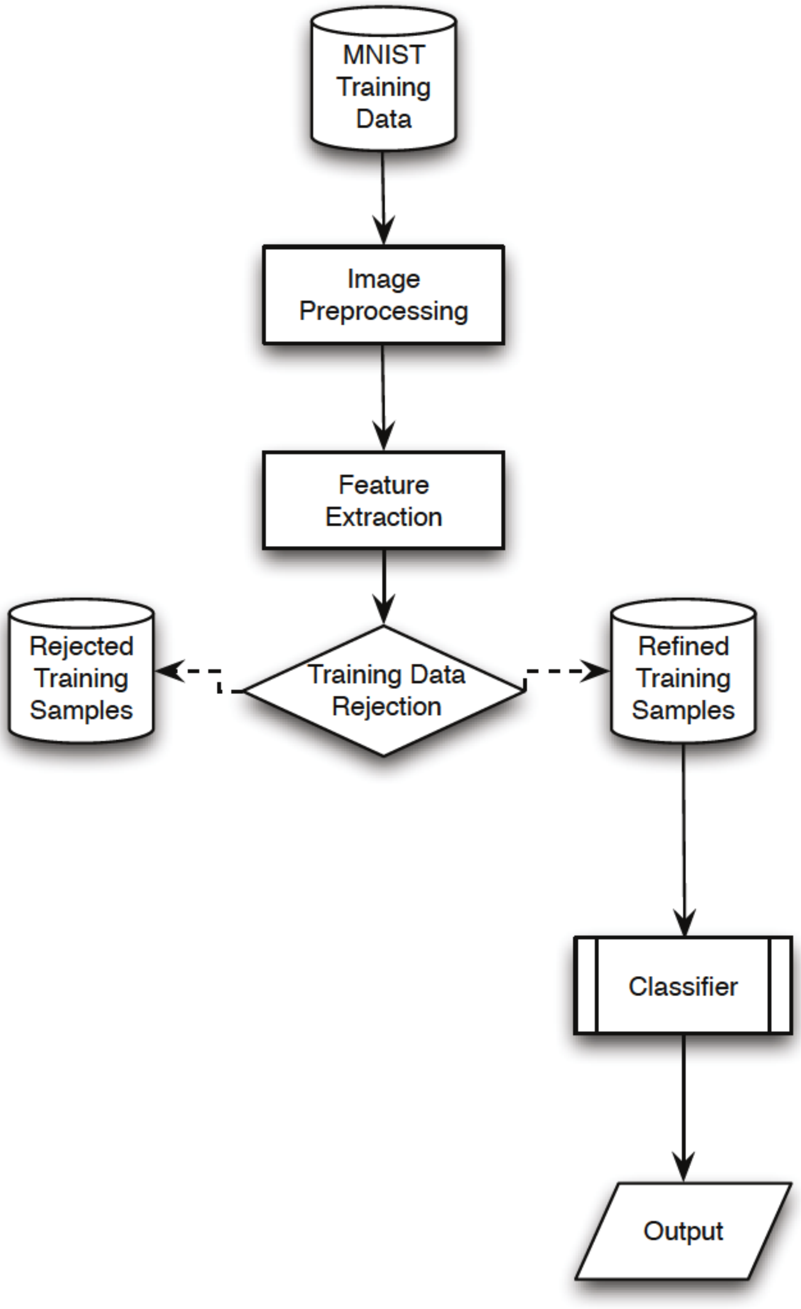


Figure 5: Overall picture of the error preventing and tolerant recognition system.

Similarly, the testing set is analyzed using statistics to find its own set of outliers. The following diagram provides a general view of the overall system that is being examined last in this section.

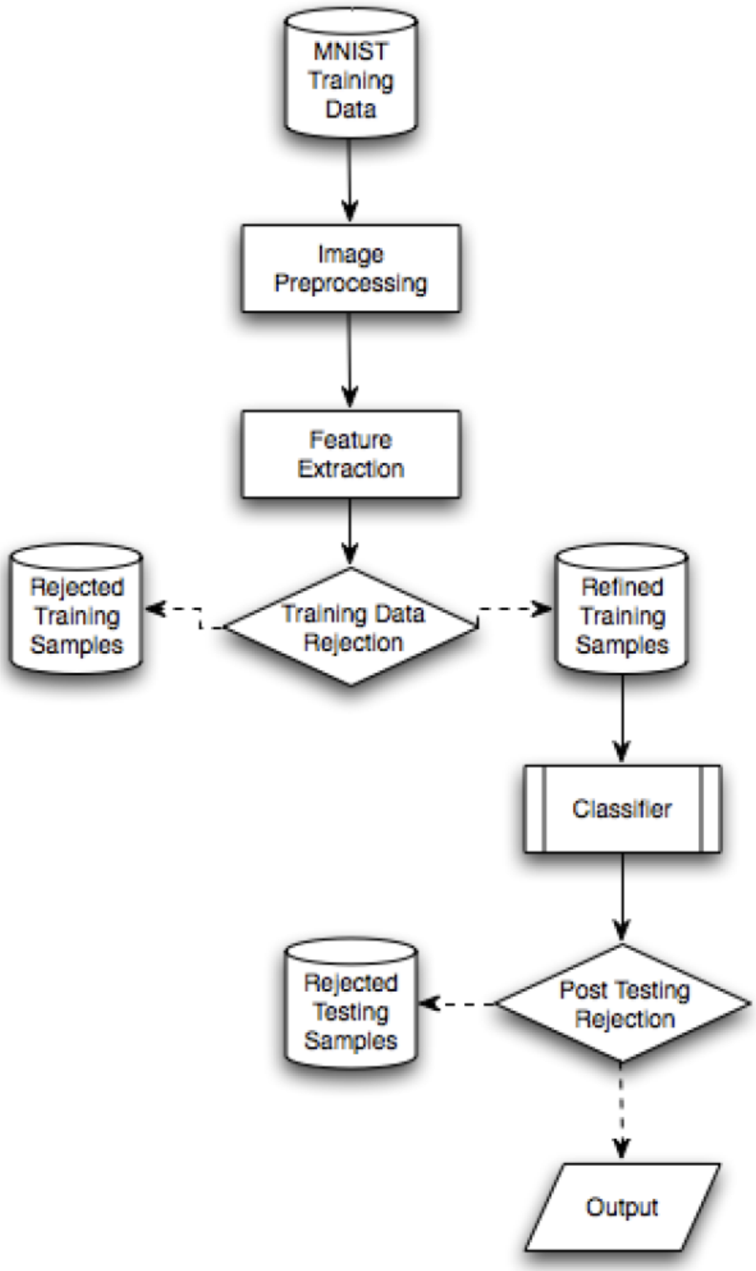


Figure 6: Overall picture of the error preventing and tolerant recognition system.

The training set consists of the 60,000 training samples from the MNIST database.

3.1.1. Image Pre-processing

It is best to apply pre-processing techniques to images as it usually yields better results than using raw images data for both training and testing processes. This procedure is usually done before feature extraction. As a result, it makes the subsequent features extracted from the images more uniform and relevant for the classifier. There are many image pre-processing techniques, such as noise removal, slant correction, normalization, skeletonization, binarization that are regularly used in pattern recognition. The steps taken for pre-processing largely depend on the features we are extracting in the next step. For example, if we are extracting gradient features, it is not necessary to perform binarization and skeletonization. In Shi's paper [16], they have started from a binary image and applied a mean filter to transform the image back to grayscale. It would be counter productive to binarize an image and then convert it back to grayscale. Moreover, it has been verified that by doing so, the classifier yielded equal or worst recognition rates. Also, after the large feature vector is extracted (refer to gradient feature section of previous chapter) it will be down sampled twice to reduce its size. We need all the information available from the image to have a feature vector that reflects the gradient of the original image. Thus, skeletonization would drop some vital information needed for the classifier at a later stage. In contrast, when we extracted the projection feature, all these steps were necessary to produce a simpler image. The feature creation for this case is a much simpler process as it works best on binary images and does not involve down sampling of the obtained feature vector.

In the following paragraph, we describe how the above pre-processing methods are being applied to the images data set.

Noise removal is performed using a 3 x 3 mean filter over the image only once. Normalization is done using the linear method. The target normalization size is 32 x 32 since it generates a better recognition rate based on the experiment by He et al. [19]. Thinning is then applied on the normalized image. The thinning algorithm used is the successful Zhang-Suen skeletonization method [4]. This method is one of the most efficient at reducing an image to its basic skeleton.

3.1.2. Projection Feature Extraction

The projection feature is extracted from both the vertical and horizontal axes. First, we sum up all the image components on the vertical axis for each row of the image. After, we sum up all the image components on the horizontal axis for each column of the image. At this stage, we have obtained two feature vectors of size equal to the image's height and width respectively. In our particular experiment, we have two feature vectors of size 32. At the end of the feature extraction process, we combined both feature vectors to obtain a vector of size 64 is generated. The feature extraction process is applied to both training and testing samples.

3.1.3. Training rejection

At this point, the data passed down for the training and testing processes takes different paths. The feature vector obtained from the training samples will be fed to a rejection system. The system separates the images into ten classes by grouping them using their respective label. Statistics are used to generate an average value for each numeral class. The rejection criterion used depends on the standard deviations of the classes. Any sample outside of a certain number of standard deviations from the mean value will be removed from the training set. This process is uniformly applied to all numeral classes simultaneously, such that a class that has very tightly corresponding samples would have fewer samples removed than one that is spread out in range.

3.1.4. Algorithm for training and testing rejection

Usually, the 60,000 training samples are used to train the classifier. Some samples in this set will be removed through the outliers' removal process described above. The same 10,000 testing samples are used for testing the performance of the classifier after being trained with the refined training set. The outliers' removal process is done by following these steps:

- Group the feature vector per numeral class
- Sum up the feature vector
- Compute the average value for the feature vector sum
- Remove the sample that has an average feature vector sum outside of three standard deviations in the training set

The following table shows each numeral's mean and standard deviation computed during the rejection process using the above algorithm on projection feature vectors.

Table 2: Mean value and std dev. computed per numeral based on the projection feature.

Numeral	Mean	Standard Deviation
0	4.790866	0.591356156
1	1.856956	0.305521736
2	4.055002	0.854580716
3	3.91357	0.729044487
4	3.471089	0.484303681
5	3.880114	0.727845774
6	3.976462	0.859012713
7	3.072418	0.494031081
8	4.908323	0.83146505
9	3.854429	0.608604491

After training the SVM classifier, it achieved an accuracy of 93.27% with a substitution rate of 6.73% on the original MNIST training set. This will serve as the baseline for comparison in the subsequent sub-sections. The above algorithm is also applied to the testing set as a post-classification rejection system in the second part of this chapter.

3.2. Results of rejecting in the training set only

The following table shows the accuracy rates, substitution rates and the number of samples rejected in the training set when different standard deviations are used. The best results are shown in bold in the following table.

Table 3: Accuracy rates, substitution rates, and samples rejected from the training set at different std dev.

Standard Deviation	Accuracy (%)	Rejected Samples	Substitution (%)
1.0	89.02	17206	10.98
1.1	90.05	14932	9.95
1.2	90.55	12364	9.45
1.3	91.07	10548	8.93
1.4	91.62	8690	8.38
1.5	91.88	7259	8.12
1.6	92.23	5867	7.77
1.7	92.54	5065	7.46
1.8	92.68	4059	7.32
1.9	92.71	3315	7.29
2.0	92.74	2836	7.26
2.1	92.71	2190	7.29
2.2	92.85	1770	7.15
2.3	92.94	1467	7.06
2.4	92.97	1213	7.03
2.5	93.09	1004	6.91
2.6	93.13	811	6.87
2.7	93.20	690	6.80
2.8	93.20	522	6.80
2.9	93.25	429	6.75
3.0	93.25	379	6.75
3.1	93.26	305	6.74
3.2	**93.28**	**274**	**6.72**
3.3	**93.29**	**237**	**6.71**
3.4	**93.29**	**205**	**6.71**
3.5	93.25	177	6.75
3.6	93.26	160	6.74
3.7	**93.28**	**148**	**6.72**
3.8	**93.30**	**119**	**6.70**
3.9	**93.30**	**106**	**6.70**
4.0	93.27	99	6.73
4.1	93.27	83	6.73
4.2	93.24	79	6.76
4.3	93.25	77	6.75
4.4	**93.28**	**67**	**6.72**
4.5	**93.29**	**65**	**6.71**

The following graphs are plotted using the data table above. It shows the trend in the accuracy rate against the standard deviation as the latter is increased. As the standard deviation increases, we keep more samples and only the most outlying samples are rejected from the training set. The best results are an improvement of 0.03% at 3.8 and 3.9 standard deviations.

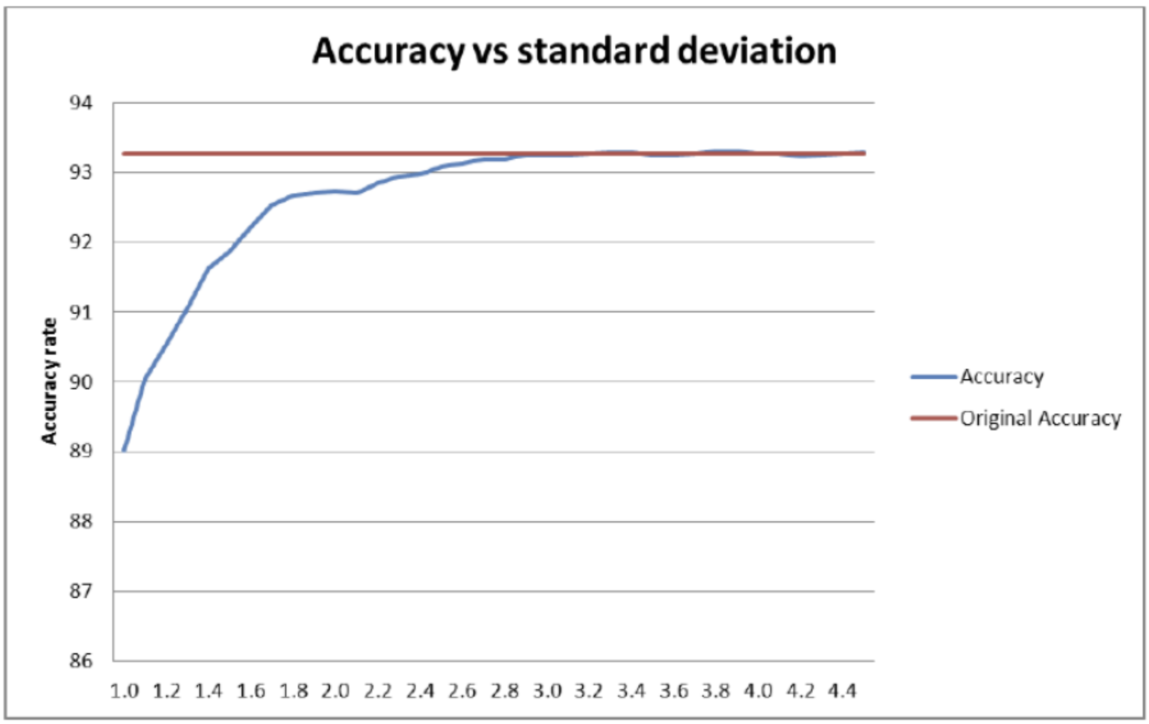


Figure 7: Accuracy against number of standard deviations used in the training sample rejection process.

This is a magnified version of the same graph on the portion of interest.

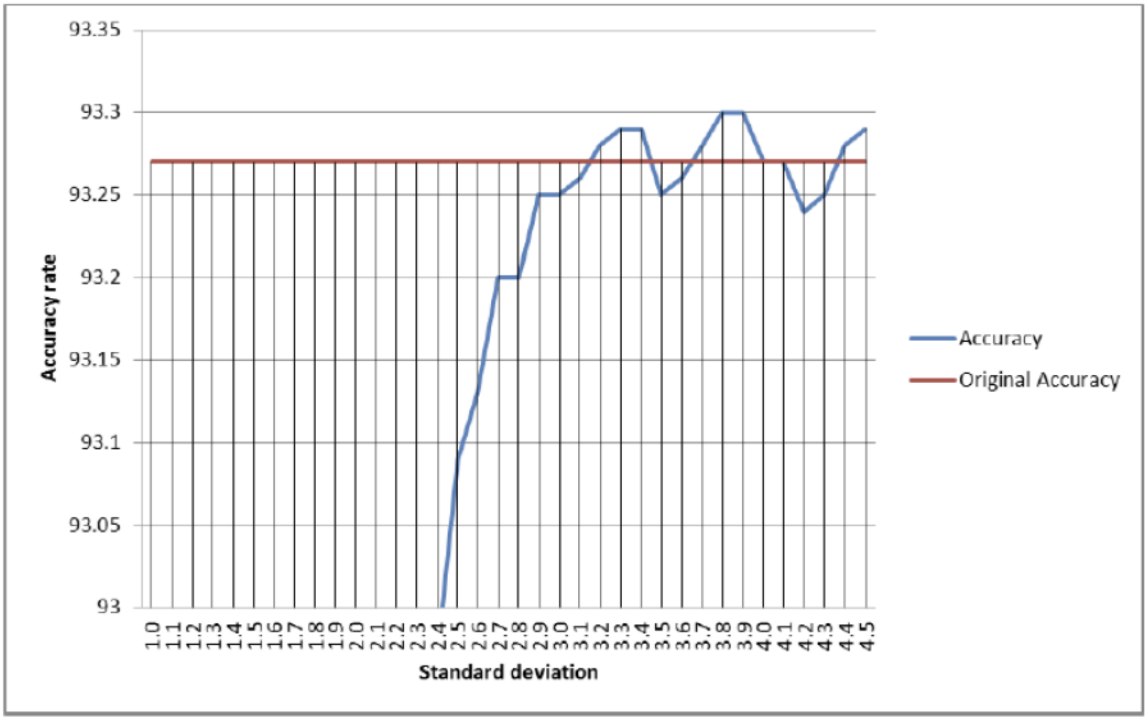


Figure 8: Magnified portion of the previous graph that is of interest.

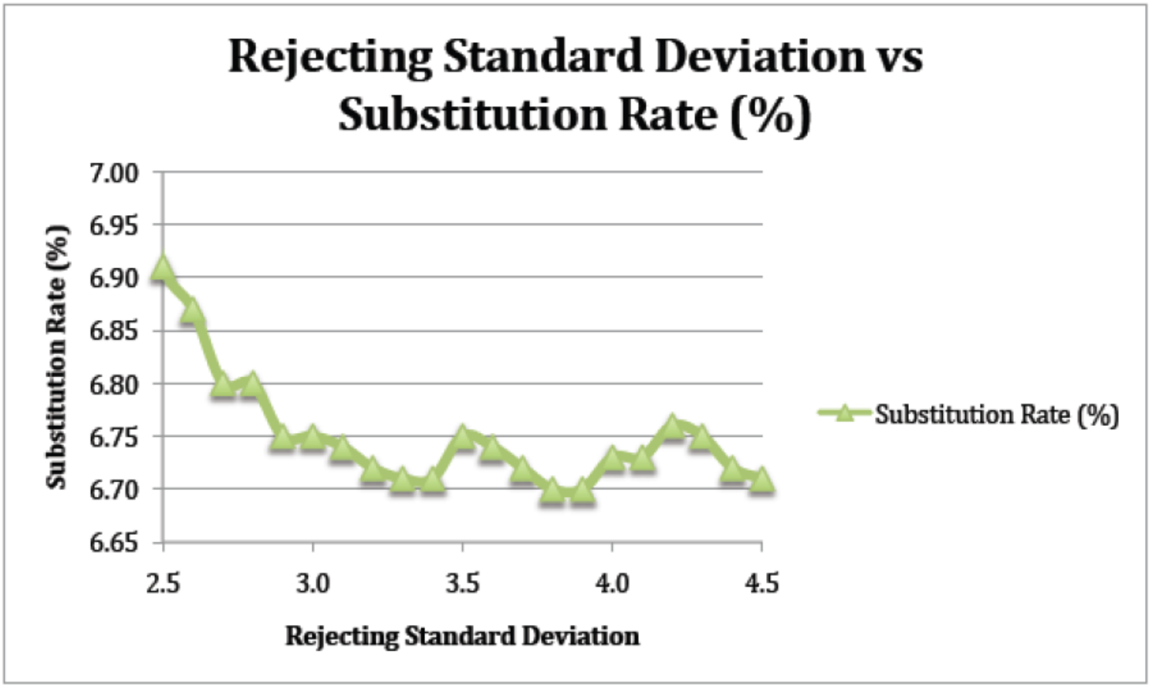


Figure 9: Standard Deviation vs Substitution Rate graph focusing on portion of interest.

3.3. Analysis of rejecting in the training set only

It was anticipated that if too many training samples were rejected, the accuracy would suffer. A classifier without sufficient training samples will misclassify testing samples due to its limited set. As we can observe from the data obtained and figures above, rejecting too many samples does decrease the classifier's effectiveness to recognize samples. Rejecting more samples with a standard deviation less than three is useless.

It can be noticed that the accuracy is very close to the original accuracy (without any modification to the training set) when only training samples outside of three standard deviations are rejected. Within the standard deviation range of 3.0 to 4.5, 19 out of 25 standard deviation values used yielded equal or better accuracy.

3.4. Evaluation of rejecting in the training set only

We can conclude that in most cases, rejecting outliers in the training set can be useful to provide better accuracy results.

While it might not be trivial to find the optimal standard deviation to use based on different feature sets and training sets, it can be observed that it is useful to eliminate mislabeled samples by making the overall classification system more robust. The rejection step before training ensures that only valid samples are passed down to the classifier's training system. The system can thus be made error tolerant and shielded from potentially unintentional errors.

3.5. Results of rejecting in the testing set only

The following table shows the accuracy rates, substitution rates and the number of samples rejected in the testing set alone when different standard deviations are used.

Table 4: Results of rejecting samples in the testing set using different std dev.

Rejecting Standard Deviation	Accuracy Rate (%)	Rejected Samples in Testing Set	Substitution Rate (%)
1.0	68.99	2689	4.12
1.1	72.55	2304	4.41
1.2	76.63	1863	4.74
1.3	79.41	1556	5.03
1.4	82.07	1263	5.30
1.5	84.00	1044	5.56
1.6	86.29	793	5.78
1.7	87.45	664	5.91
1.8	88.82	512	6.06
1.9	89.96	384	6.20
2.0	90.55	320	6.25
2.1	91.27	238	6.35
2.2	91.79	179	6.42
2.3	92.17	133	6.50
2.4	92.52	90	6.58
2.5	92.66	73	6.61
2.6	92.78	59	6.63
2.7	92.88	47	6.65
2.8	93.03	31	6.66
2.9	93.10	23	6.67
3.0	93.13	20	6.67
3.1	93.18	15	6.67
3.2	93.20	10	6.70
3.3	93.20	10	6.70
3.4	93.20	10	6.70
3.5	93.20	10	6.70
3.6	93.20	9	6.71
3.7	93.21	8	6.71
3.8	93.22	7	6.71
3.9	93.23	6	6.71
4.0	93.23	6	6.71
4.1	93.23	5	6.72
4.2	93.24	4	6.72
4.3	93.24	4	6.72
4.4	93.25	3	6.72
4.5	93.26	2	6.72

The next figure plots the results from the above table as the standard deviation against the substitution rate. A linear plot of the original substitution rate without any rejection is included for comparison.

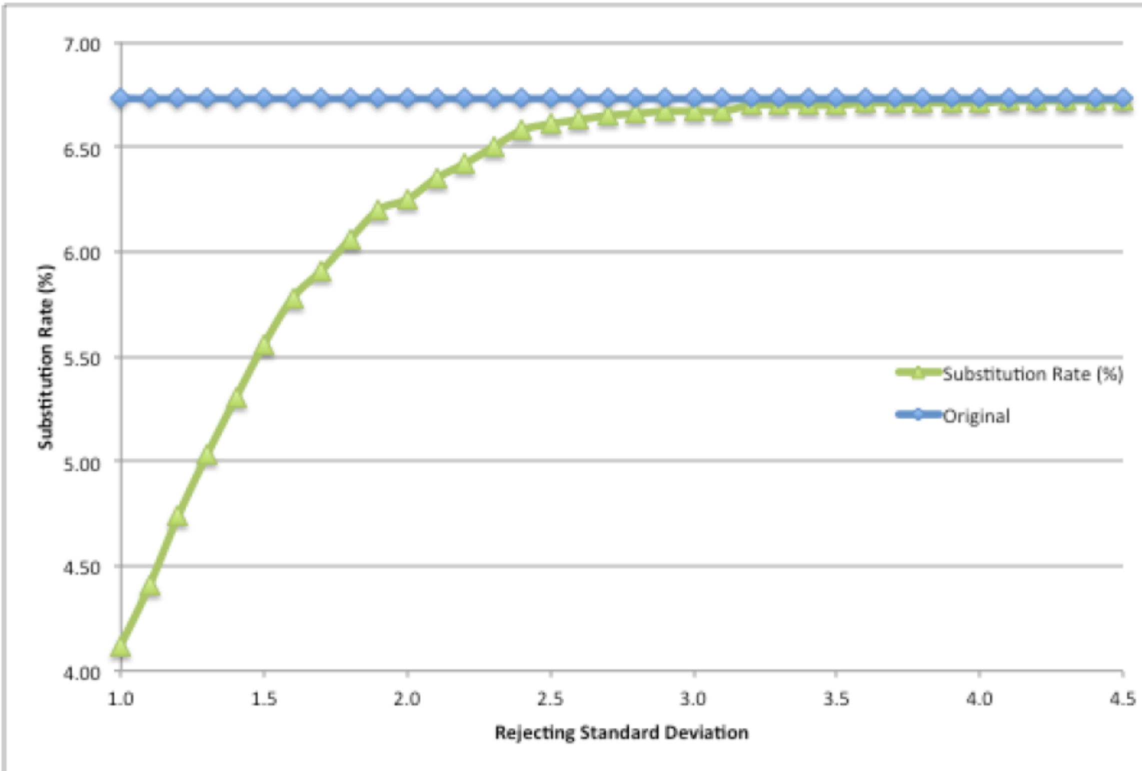


Figure 10: Graph of rejecting standard deviation against substitution rate in the testing set only

3.6. Analysis of rejecting in the testing set only

We observe a sustained drop in the substitution rate as more samples are rejected from the testing set. This behavior was expected. However, the accuracy is also taking a toll as misrecognized samples and correctly recognized samples are both growing at about the same pace in the rejected set. There is a slight decline of both accuracy and substitution rate in the standard deviation ranging from 3.0 to 4.5.

From 2.4 to 3.0, the rate of decrease starts to accelerate. Significant drop in accuracy and substitution rates are observed from 2.3 onward.

3.7. Results of rejecting in both testing and training set

The full table, which shows all the results obtained combining a variation in standard deviation for rejection in the testing and training sets, can be found in Appendix A. The following graphs from figures 11 to 15 provide a better mean of communicating the results.

These five charts plot the results of rejection in both testing and training sets from the table in Appendix A. The first chart shows the trend with different standard deviations where results are lower than expected. The second, third, fourth, and fifth charts emphasize more on the area of interest. All charts demonstrate the substitution rate against the different standard deviations used for rejection in the testing set. Each line represents a specific standard deviation used for rejection in the training set.

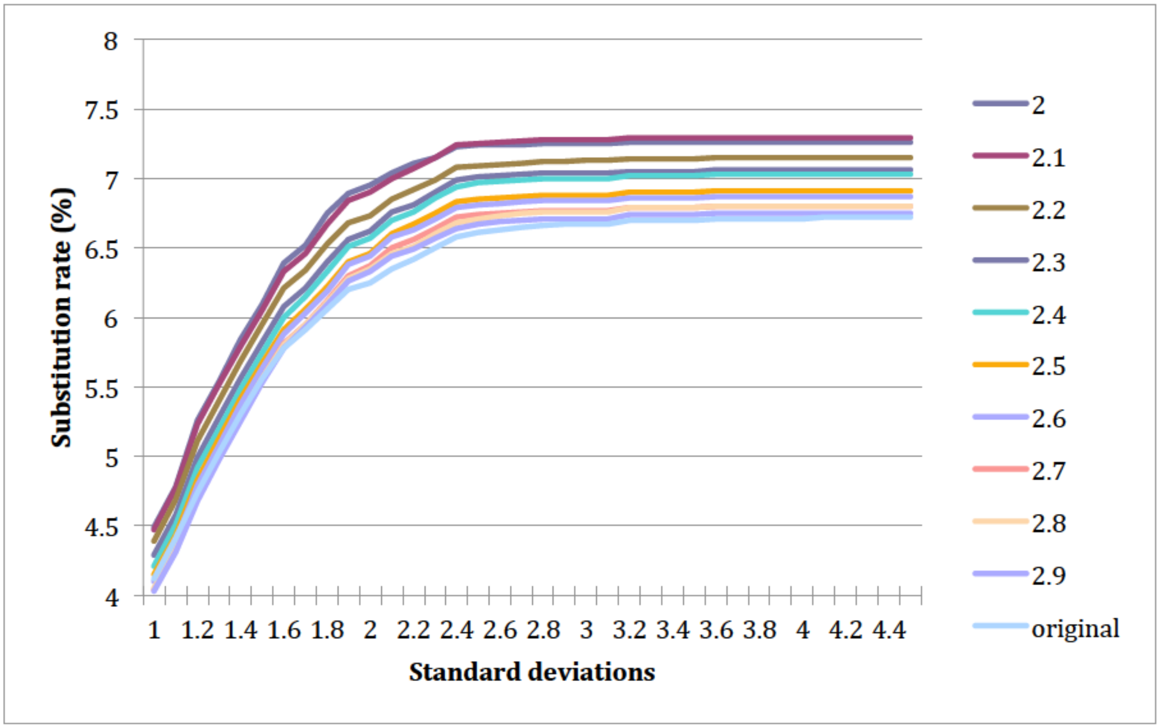


Figure 11: Graph of rejecting standard deviation (2.0-2.9) against substitution rate in both training and testing sets.

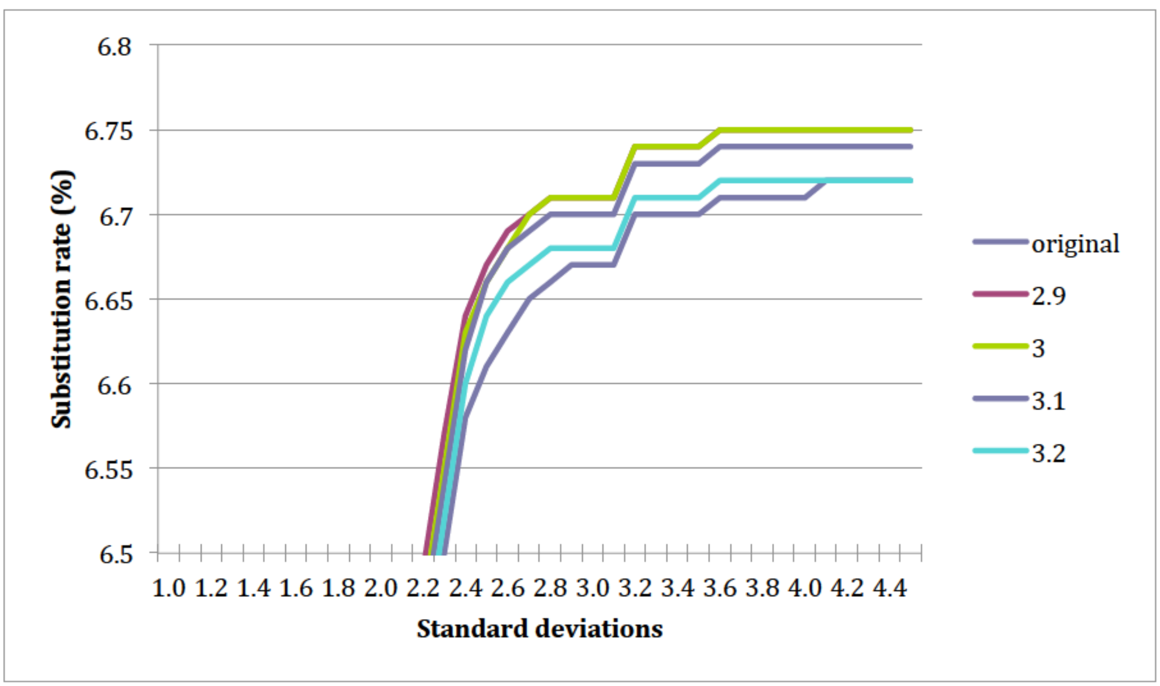


Figure 12: Graph of rejecting standard deviation (2.9-3.2) against substitution rate in both training and testing sets.

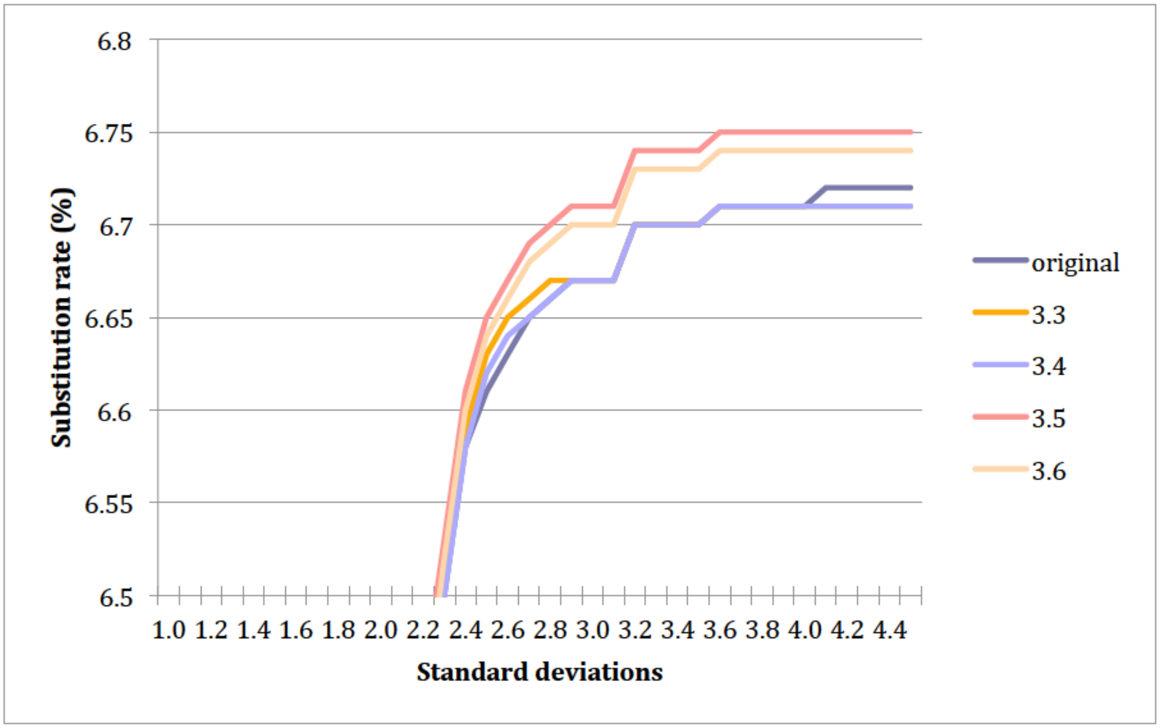


Figure 13: Graph of rejecting standard deviation (3.3-3.6) against substitution rate in both training and testing sets.

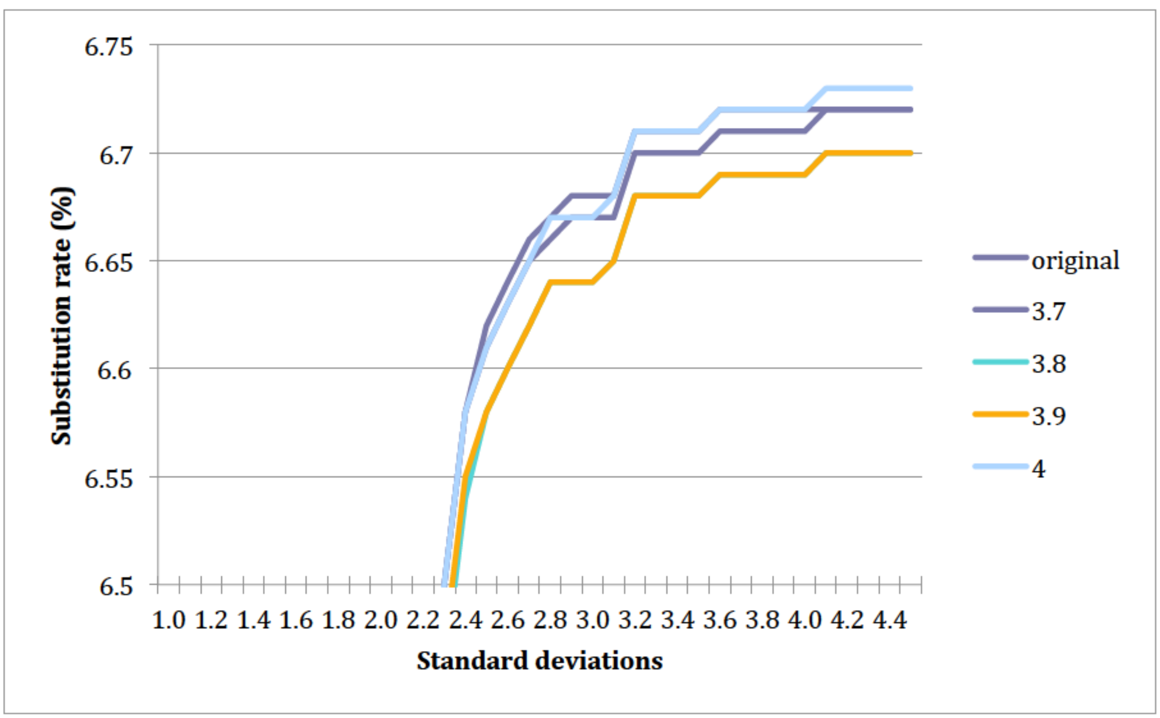


Figure 14: Graph of rejecting standard deviation (3.7-4.0) against substitution rate in both training and testing sets.

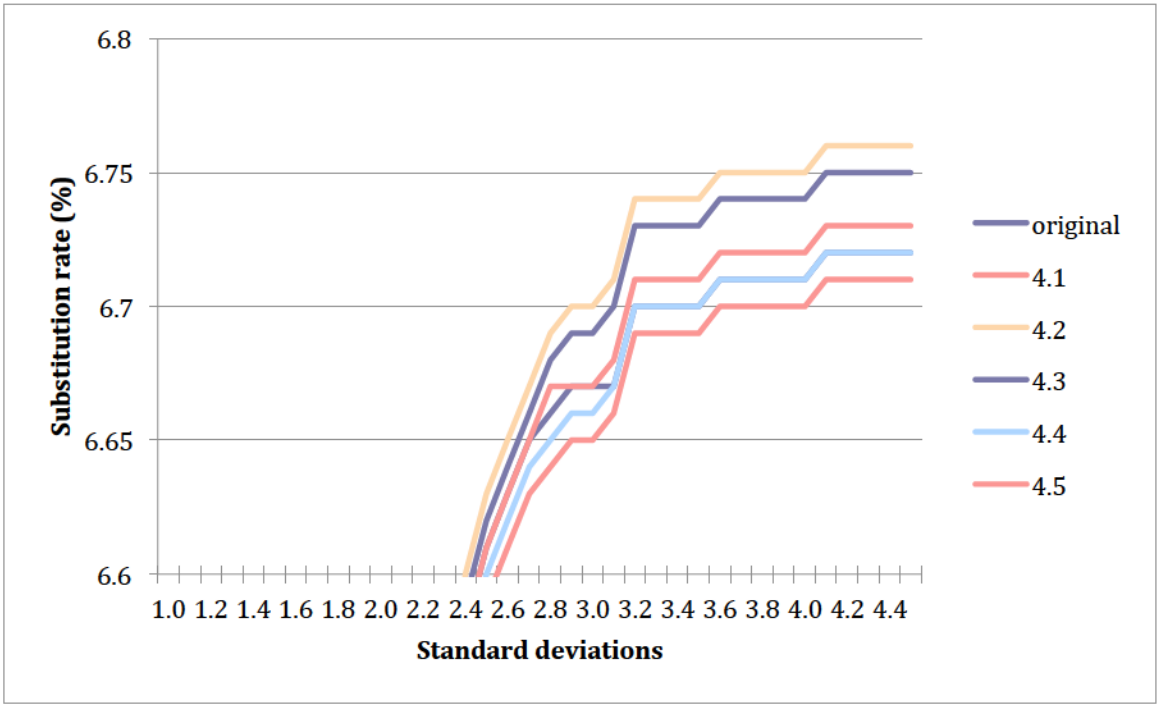


Figure 15: Graph of rejecting standard deviation (4.1-4.5) against substitution rate in both training and testing sets.

The original reference line plots the data obtained for no rejection in the training set and rejection only in the testing set. From the above graph, it can be observed that the general trend for rejection in both training and testing set is similar to the trend for rejection in only the testing set. From left to right on the graphs, the substitution rate decreases slightly and the rate of decrease accelerates with a significant drop in both accuracy and substitution rates starting mid-way.

3.8. Analysis of rejecting in both testing and training set

It was anticipated that if samples were rejected in the training set, the accuracy might suffer. This effect is amplified when samples are rejected from the testing set. It is inevitable that some of the correctly recognized samples will be discarded along with the outliers. It is even more important that we reject only the real outliers in both the training and testing sets. Choosing a relatively large value for the rejection criteria or the standard deviation, can limit the numbers of good samples being rejected.

As observed in the previous experiment, rejection in the standard deviation range of 3.0 to 4.5 in the training set yielded the best results. Likewise, the same is observed for rejection in the standard deviation range of 3.0 to 4.5 in the testing set. For this range, less than ten samples are rejected in the testing set and as a result, the accuracy rate does not change significantly.

3.9. Evaluation of rejecting in both testing and training set

We can conclude that in most cases, rejecting outliers in the training set and testing can be useful to provide better accuracy rates and lower substitution rates.

It is observed that by eliminating mislabeled samples and rejecting some outlying samples from the testing set, there is no possible improvement in accuracy. Accuracy can only decrease, as correctly predicted samples will eventually get rejected along the outliers. However, by eliminating the outliers outside four standard deviations, we obtained accuracies that are very close to the original value with a slightly lower substitution rate. Also, our traditional rejection system can be improved with a more sophisticated approach and better results can be noticeable in terms of recognition and substitution rates. In the presence of mislabeled samples for training, the rejection system would detect these as outliers and the latter would not reach the training stage of the classifier. At the same time, samples that are not strongly correlated to the predicted class should be rejected from the testing set to prevent errors. The system can be made error tolerant by discarding samples in the training set and the error prevention mechanism works by rejecting outlying predicted samples from the testing set.

Chapter 4: Rejection in Training Set on Gradient Features

In this chapter, a rejection in the training set for gradient features using a SVM based classification system is proposed for handwritten numerals recognition.

The hypothesis is that different features would vary the behavior in the rejection mechanism and impact the classifier's accuracy. Also, the gradient features are known to provide the classifier with a feature vector achieving one of the highest accuracies to date [16].

In the current chapter, we would detect and reject offending samples from the training set based on the gradient features to achieve a more reliable recognition system.

4.1. Methodology

4.1.1. Image Pre-processing

The training set rejecting classification system uses features statistics to identify outliers. This method is exactly the same as in the previous two chapters. We first apply image pre-processing techniques to the 60,000 training samples from the MNIST database. Noise removal, slant correction, normalization, skeletonization and binarization are applied respectively to the training set in the previous chapters. However, since we are using the gradient features, which are extracted from gray-scaled image, the binarization and skeletonization steps are skipped in this chapter. It has been verified that the classifier yielded better accuracy without these two extra image pre-processing techniques.

4.1.2. Gradient Feature Extraction

The gradient feature is implemented following the procedure described in the paper published by Shi et al [16].

The gradient feature is obtained from a gray-scaled image. A feature vector of 400 features is obtained at the end of this process.

4.1.3. Training rejection

The extracted feature vectors for the classifier's training process and testing process go through different routes at this point. The feature vector obtained from the training samples are passed to the rejection system. The system classifies each feature vector into its respective class using its label. The aggregated feature vectors are then used to generate an average value for each numeral class. We then compute the statistical information such as the mean and standard deviation for each individual class. The rejection criterion depends on the intra class standard deviations. The rejection mechanism discards any sample outside of a specified number of standard deviations from each class' mean value. This process is uniformly applied to all numeral classes simultaneously.

4.1.4. Algorithm for training rejection

Again, not all 60,000 training samples are used for training the classifier. Some samples from the training set will be discarded through the outliers' removal process described in the previous section. The 10,000 testing samples are unaltered and used for testing the performance of the classifier after being trained with the refined training set. The outliers' removal process is done in the same way:

- Group the feature vector per numeral class
- Sum up the feature vector
- Compute the average value for the feature vector sum
- Remove the sample that has an average feature vector sum outside of three standard deviations in the training set

After training the SVM classifier, it achieved an accuracy of 98.97% with a substitution rate of 1.03% on the original 60,000 samples MNIST training set. This will serve as the baseline for comparison in the subsequent sub-sections where training set rejection is applied.

The following table shows the accuracy rates, substitution rates, and the number of samples rejected in the training set when different standard deviations are used. The best results have been highlighted from the following table.

Table 5: Accuracy rates, substitution rates, and number of rejected samples from the training set at different std dev.

Rejecting Standard Deviation	Accuracy Rate (%)	Rejected Samples in Training Set	Substitution Rate (%)
2.5	98.78	841	1.22
2.6	98.81	709	1.19
2.7	98.86	576	1.14
2.8	98.86	486	1.14
2.9	98.84	397	1.16
3.0	98.89	335	1.11
3.1	98.88	283	1.12
3.2	98.87	238	1.13
3.3	98.90	199	1.10
3.4	98.91	179	1.09
3.5	98.90	160	1.10
3.6	98.92	147	1.08
3.7	98.90	128	1.10
3.8	98.93	114	1.07
3.9	98.96	102	1.04
4.0	98.94	93	1.06
4.1	**98.97**	**83**	**1.03**
4.2	98.96	78	1.04
4.3	98.93	73	1.07
4.4	98.96	68	1.04
4.5	**98.97**	**62**	**1.03**

The following graphs are plotted using the data table above and show the trend in the accuracy rate against the standard deviation as the standard deviation used increases. As the standard deviation increases, fewer samples are removed and only the most outlying samples are rejected from the training set. The best results are at 4.1 and 4.5 standard deviations.

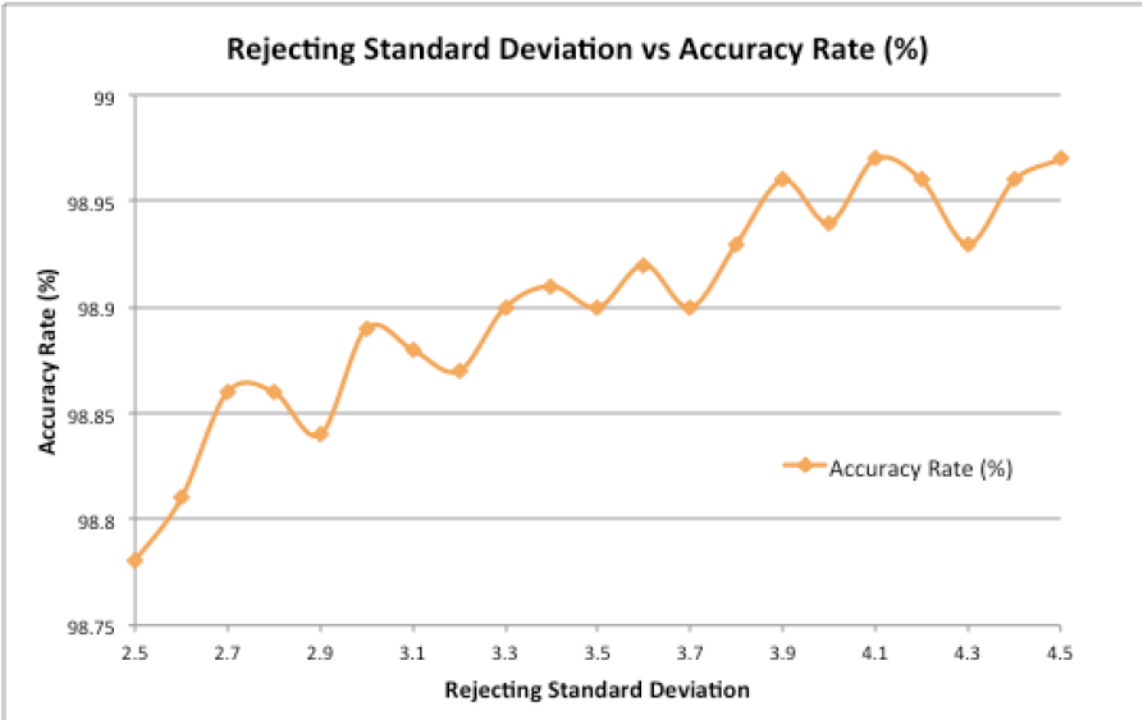


Figure 16: Accuracy against standard deviation used in the training sample rejection process.

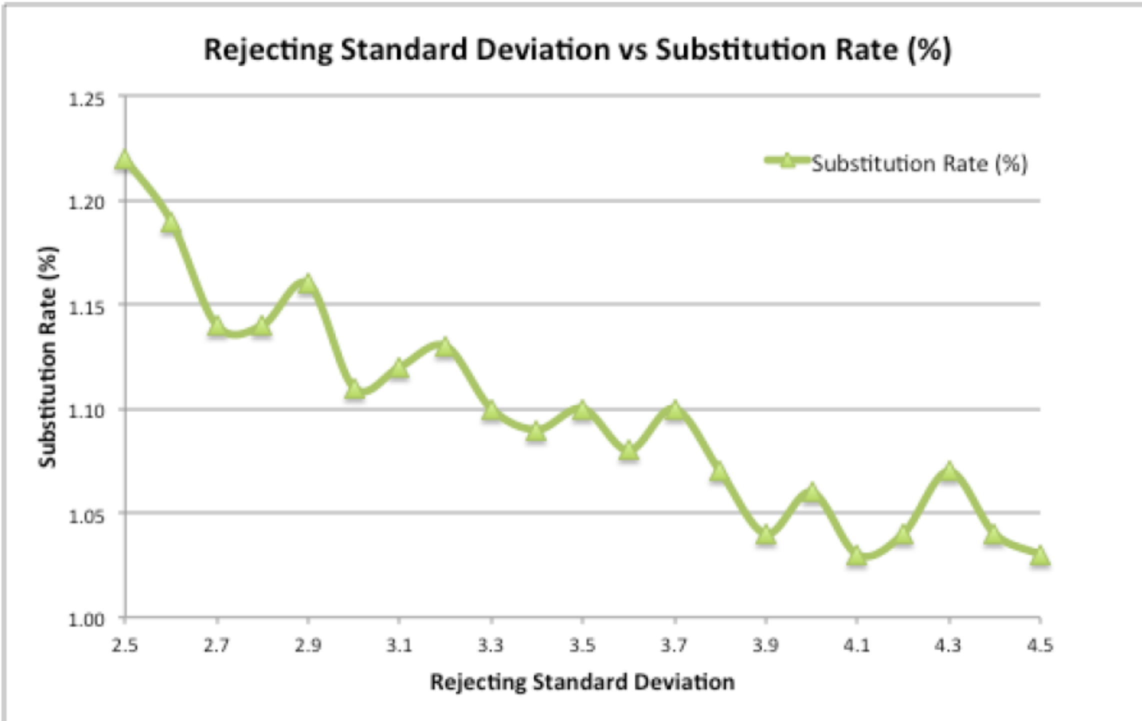


Figure 17: Substitution rate against standard deviation used in the training sample rejection process.

4.2. Analysis

The training is done using (60,000 – outliers) samples from the MNIST training set. The testing is done on the standard 10,000 MNIST testing set. The original accuracy without any rejection is 98.97%.

The above table shows the results obtained for the gradient feature using libSVM with parameters c =32 and g =0.0078125. Each row represents the results for a different rejection criterion applied to the standard 60,000 MNIST training set.

It was expected that the accuracy would drop significantly if too many samples were discarded from the training samples set. The classifier would be skip useful samples during the training process and the testing process would be more prone to errors.

As we can observe from the table, the results are close to the original for a rejection range greater than 3.9 standard deviations. However, there are no observable improvements. In contrast to the projection feature tested in earlier chapters, there were improvements when the standard deviation used was greater than 3.7 standard deviations.

There are many reasons explaining our results with the gradient feature. First, this feature is very accurate and a lot of useful information is extracted. Removing samples from such a feature set would strip the classifier of relevant samples. However, as efficient as this feature can be, there ought to be bad samples in the training set, which can be removed without affecting our classifier's accuracy rate.

Another explanation is that some useful samples are being discarded along with these bad samples. As a result, it cancels or overrides any observable improvement. To confirm this hypothesis, an examination of the rejected samples is done in the following section.

4.3. Classification of Rejected Samples from the Training Set

In this section, we analyze the samples that are rejected from the training set using different standard deviations. The training samples have been classified into different categories. Some of the categories are derived from Tan's paper [12]. The samples have been categorized into three fundamental groups: legible samples, unrecognizable samples, and confusing pairs samples. The legible group is further refined into: good samples, poor samples, thick stroke samples, and very slanted samples. Below is a brief description of each individual category mentioned above ranked by their usefulness to the classifier's training set. The first category is of the highest relevance, whereas the last category is the most irrelevant in the training set.

1. The good samples class is composed of images that can be classified accurately to their respective labeled class easily. These samples are still desirable in the training set.

2. Very slanted samples are numerals that are written in a very inclined or distorted way. The samples can be moderately challenging to identify but are in general of better quality than the thick stroke or poor samples.

3. Thick stroke samples contain numerals that are written using a peculiarly wide writing instrument when compared with others samples. Some structural features may not be obviously identifiable and even lacking, such as loops and intersection points.

4. Poor samples encompass images that are challenging to classify. These samples are usually noisy or partially damaged or clamped.

5. The confusing pairs samples consist of images that can be classified as more than one class and are difficult to attribute to one particular class. It is arguable which class they belong to without looking at their respective labels.

6. The unrecognizable samples category contains samples that even humans cannot identity with confidence.

Any sample that falls into a category of equal or lower ranking than three on our scale should be discarded from the training set. Good and very slanted samples can be kept in our training set to maintain or improve the classifier's accuracy.

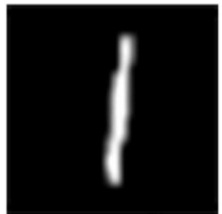
The following are the classified samples in each category for different standard deviations used in the rejection mechanism above. As the standard deviation used becomes smaller, more samples are identified as outliers and removed from the training set. All samples discarded at a particular standard deviation are inclusive of the preceding sets. Each group of samples shown previously to the current group is a subset of the current group. For instance, the samples discarded at standard deviation 3.5 include all the ones shown at standard deviations 2.5, 3.0, and 3.5. Each rejected outlier set has their samples classified using the categories mentioned above and in the same order of quality ranking. Their corresponding labeled number in the MNIST database is shown.

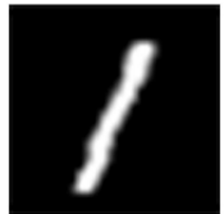
4.3.1. Samples rejected at a standard deviation 4.5

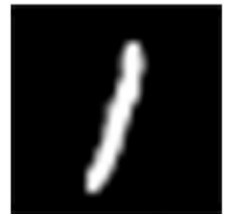
Good Samples


13885.tiff


15346.tiff


17857.tiff


1832.tiff


19290.tiff


19545.tiff

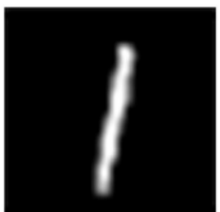

22348.tiff


23909.tiff


24959.tiff


25752.tiff


26175.tiff


28805.tiff

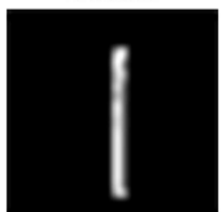

29568.tiff


29669.tiff


32613.tiff

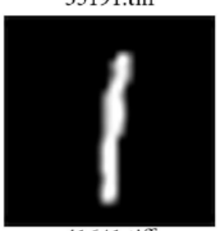

33191.tiff

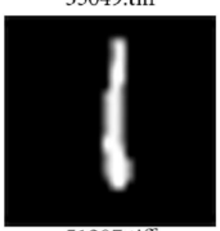

35049.tiff

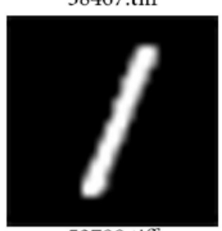

38467.tiff

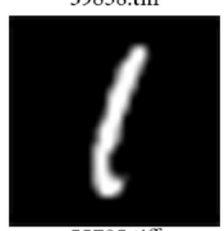

39858.tiff

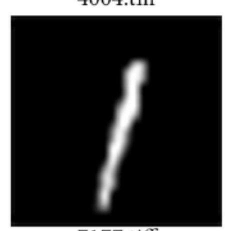

4004.tiff

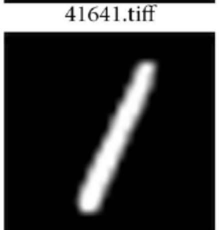

41641.tiff


51207.tiff


53788.tiff


55785.tiff


7177.tiff

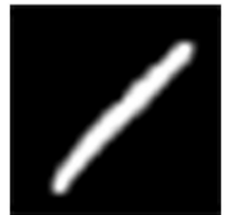

9012.tiff

Very Slanted Samples


24406.tiff


44080.tiff


50494.tiff


534.tiff


59955.tiff


8726.tiff

Poor Samples


28489.tiff


29977.tiff

Thick Stroke Samples


13339.tiff


17379.tiff


22094.tiff


24536.tiff


26945.tiff


27832.tiff


27887.tiff


29870.tiff


33092.tiff


34639.tiff


36595.tiff


38539.tiff


39078.tiff

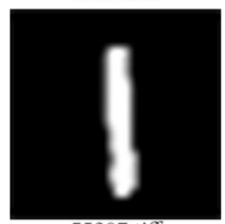

48336.tiff


48482.tiff

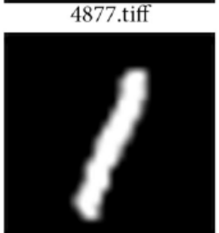

4877.tiff

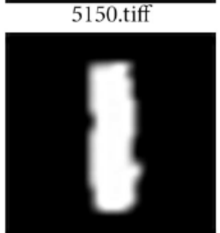

5150.tiff


52417.tiff


55307.tiff


57518.tiff


59698.tiff


6860.tiff


8824.tiff

Confusing Pairs


38088.tiff


39001.tiff

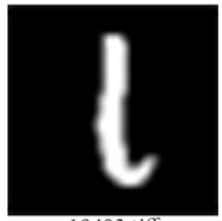

40782.tiff


5804.tiff

Unrecognizable Samples


12376.tiff


18149.tiff


18483.tiff


37117.tiff


39426.tiff

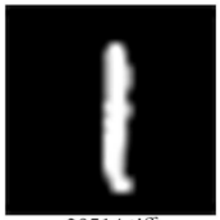

44583.tiff


50897.tiff

4.3.2. Samples rejected at a standard deviation 4.0

Good Samples


15225.tiff


28514.tiff

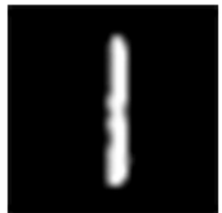

31809.tiff


3715.tiff


40328.tiff


45995.tiff


48485.tiff


52043.tiff


54774.tiff


55068.tiff

Very Slanted Samples


22268.tiff


38560.tiff


536.tiff

Poor Samples


19797.tiff


27910.tiff

Thick Stroke Samples


11246.tiff


1378.tiff


17224.tiff


39076.tiff


43333.tiff


58524.tiff

Confusing Pairs Samples


43352.tiff


49240.tiff

Unrecognizable Samples


22295.tiff


54039.tiff

4.3.3. Samples rejected at a standard deviation 3.5

Good Samples


11203.tiff


13454.tiff


13463.tiff


17032.tiff

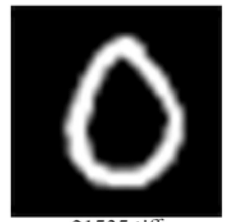

17700.tiff


17756.tiff


19310.tiff


19410.tiff


20534.tiff

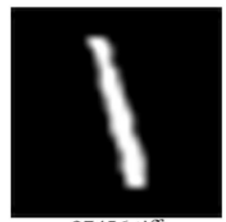

21535.tiff


2256.tiff


23309.tiff


23625.tiff


24230.tiff


27456.tiff


27790.tiff

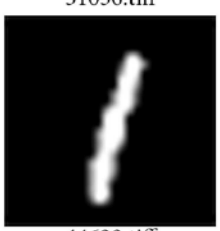

31056.tiff

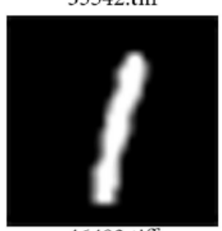

35542.tiff

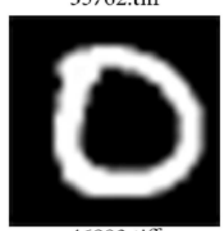

35762.tiff

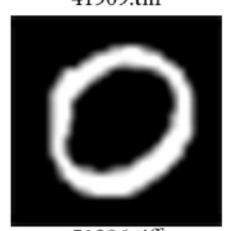

41909.tiff

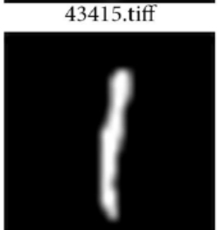

43415.tiff

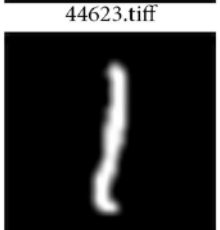

44623.tiff

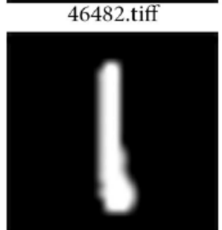

46482.tiff

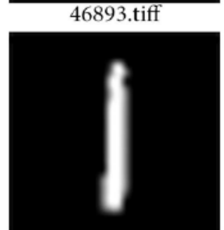

46893.tiff

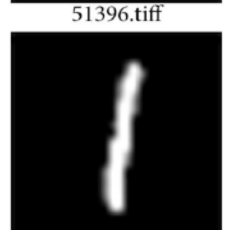

51396.tiff


51673.tiff


52235.tiff


52249.tiff


53727.tiff


55143.tiff


58581.tiff


9458.tiff

Very Slanted Samples


1444.tiff


32494.tiff


35068.tiff


43470.tiff


45988.tiff


51632.tiff


59903.tiff


7692.tiff

Poor Samples


20725.tiff


39994.tiff


50892.tiff


57021.tiff

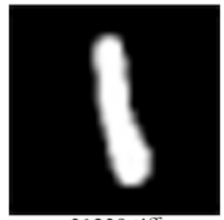

623.tiff

Thick Stroke Samples


16204.tiff


19505.tiff


20610.tiff


31239.tiff


34338.tiff


39282.tiff


41910.tiff


7534.tiff

Confusing Pairs Samples


15476.tiff


17806.tiff


17862.tiff


20334.tiff


36187.tiff


38795.tiff


50330.tiff


56217.tiff

Unrecognizable Samples


20049.tiff


24799.tiff


33507.tiff

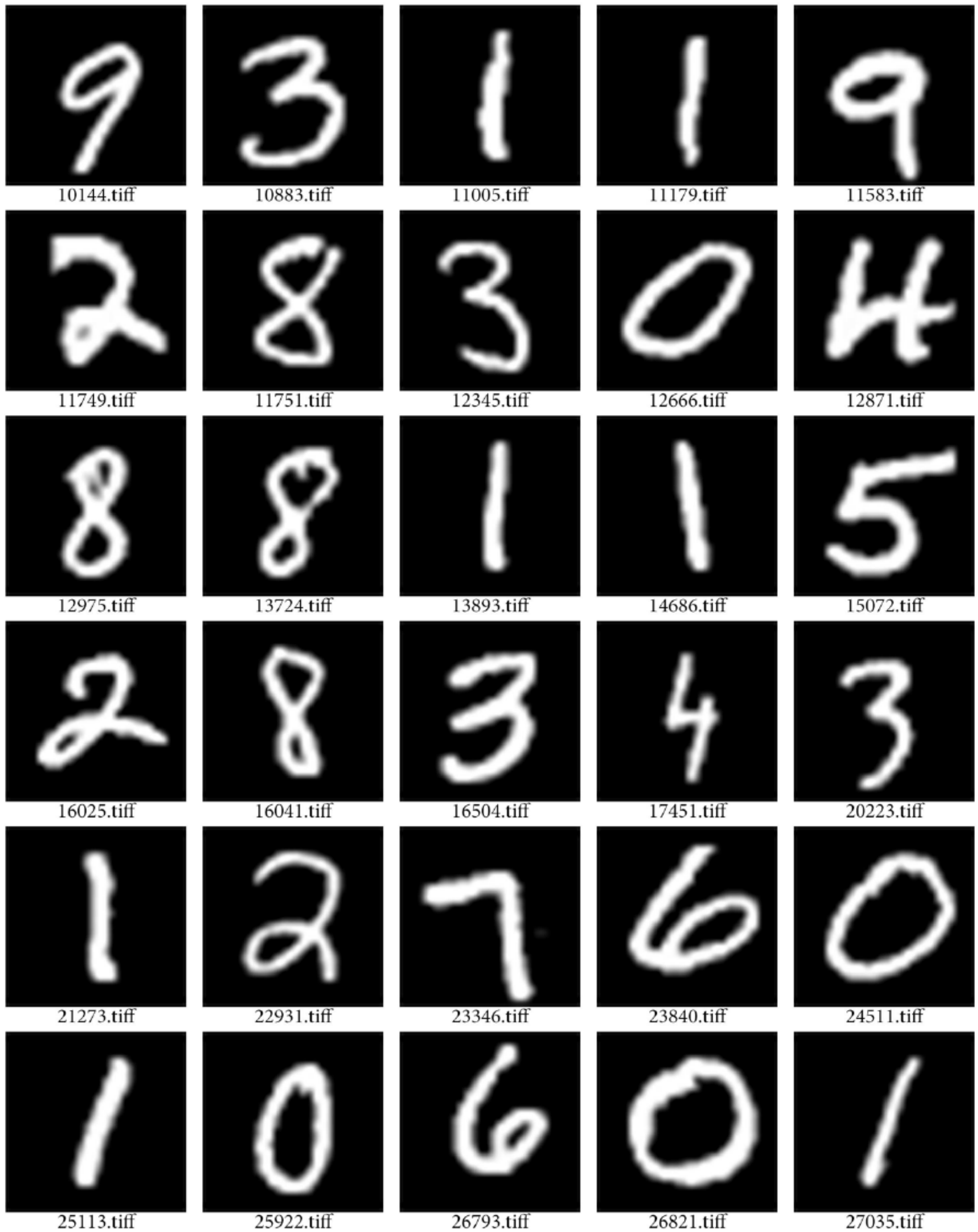

50048.tiff


8703.tiff


9718.tiff

4.3.4. Samples rejected at a standard deviation 3.0

Good Samples




2707.tiff

29333.tiff

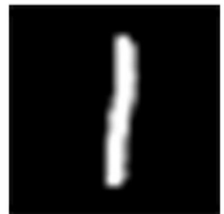
29807.tiff

29914.tiff

30014.tiff

30097.tiff

30224.tiff

30242.tiff

30338.tiff

30635.tiff

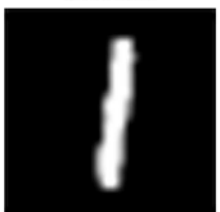
31890.tiff

32604.tiff

32764.tiff

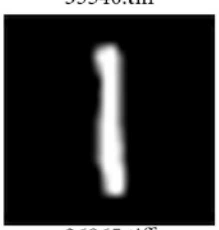
32925.tiff

33288.tiff

33540.tiff

34424.tiff

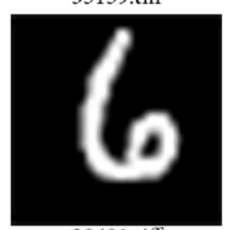
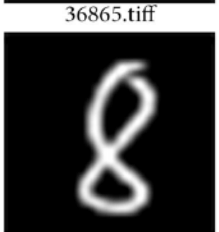
34590.tiff

34912.tiff

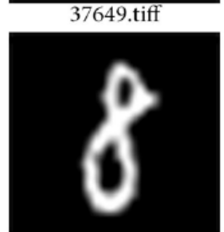
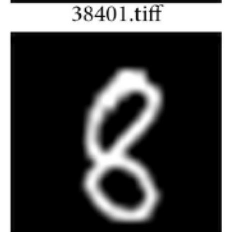
35159.tiff

36865.tiff

37528.tiff

37580.tiff

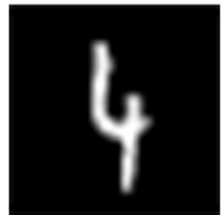
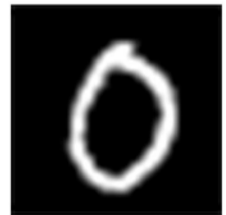
37649.tiff

38401.tiff

38957.tiff

39086.tiff

39188.tiff

39309.tiff

39351.tiff


39993.tiff

40905.tiff

41062.tiff

41193.tiff

42897.tiff

44013.tiff

44369.tiff

45137.tiff

45258.tiff

46897.tiff

47214.tiff

47786.tiff

47905.tiff

47987.tiff

48175.tiff

48418.tiff

49696.tiff

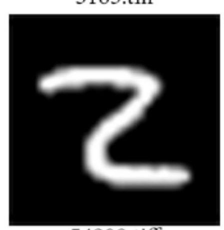
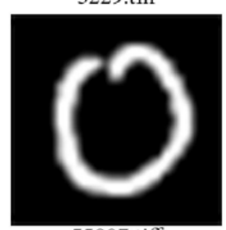
50385.tiff

5183.tiff

5229.tiff

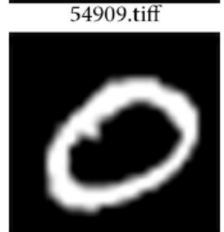
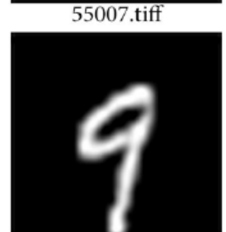
52496.tiff

539.tiff

54542.tiff

54909.tiff

55007.tiff

55332.tiff

55463.tiff

55934.tiff

56006.tiff

56258.tiff


56386.tiff

57513.tiff

58391.tiff

58646.tiff

6046.tiff

6131.tiff

6177.tiff

6489.tiff

8719.tiff

9943.tiff

Very Slanted Samples


10720.tiff

19906.tiff

32536.tiff

37874.tiff

44184.tiff

528.tiff

Poor Samples


10465.tiff


12187.tiff


13702.tiff


14322.tiff


1481.tiff


22466.tiff


22760.tiff


24318.tiff


25016.tiff


26096.tiff


29207.tiff


3046.tiff


33171.tiff


39278.tiff


39374.tiff


44163.tiff


44873.tiff

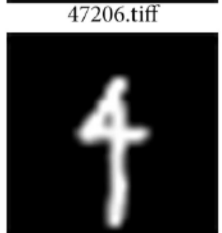

47206.tiff


48141.tiff


48312.tiff


5536.tiff


56502.tiff


5937.tiff

Thick Stroke


11111.tiff


18207.tiff


18214.tiff


2196.tiff


24035.tiff


26235.tiff


33845.tiff


36893.tiff


37762.tiff


37768.tiff


3812.tiff


40187.tiff


42158.tiff


42720.tiff


44442.tiff


48282.tiff


48720.tiff


6142.tiff


8064.tiff


8683.tiff

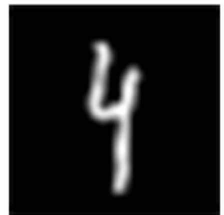

9708.tiff

Confusing Pairs Samples


13879.tiff


14441.tiff


2138.tiff


40289.tiff


40312.tiff


43822.tiff


45526.tiff


46535.tiff


48563.tiff


50392.tiff


54779.tiff


55520.tiff


56381.tiff


6811.tiff

Unrecognizable Samples


11853.tiff


13651.tiff


24182.tiff


263.tiff


30691.tiff


37397.tiff


38371.tiff


48461.tiff


53502.tiff


56453.tiff


57077.tiff

4.3.5. Evaluation of samples rejected

After comparing the samples by standard deviations, it is interesting to find the amount of undesirable training samples against the amount of desirable training samples for each rejection criterion. As stated above, any sample with a category ranking equal or better than three is considered desirable and any sample with a category ranking lesser is considered undesirable in the training set. The tables on the next page show the total of desirable and undesirable samples for the different rejection criterion used.

Table 6: Compilation of desirable samples from the rejected set at different std dev.

Standard deviation	1. Good	2. Very Slanted	3. Thick Stroke	Desirable
3.0	165	29	59	253
3.5	69	19	37	125
4.0	37	10	29	76
4.5	27	8	23	58

Table 7: Compilation of undesirable samples from the rejected set at different std dev.

Standard deviation	4. Poor	5. Confusing Pairs	6. Unrecognizable	Undesirable
3.0	37	22	23	82
3.5	13	9	13	35
4.0	7	3	7	17
4.5	4	1	5	10

To compare the number of desirable samples against the number of undesirable samples at the different standard deviations, a table with their respective percentage is computed below.

Table 8: Calculated percentage of desirable against undesirable samples at different std dev.

Standard deviation	Desirable(%)	Undesirable(%)
3.0	75.52%	24.48%
3.5	78.13%	21.88%
4.0	81.72%	18.28%
4.5	85.29%	14.71%

By analyzing the above table, our original hypothesis can be confirmed. The amount of desirable samples had always outweighed the undesirable samples. The percentage grows as more samples are rejected outside a smaller standard deviation value used for our rejection criterion. Initially at standard deviation 4.5, we are almost throwing away one good sample for each bad sample. Thus, at this point, the classifier was performing almost on par with a classifier using the full training samples set. When we reach a rejection criterion using a standard deviation of 3.0, 3 out of 4 samples that are discarded are actually desirable. The classifier does not perform well when the percentage of desirable samples is more than double the undesirable samples in the rejected samples set. It makes sense as samples that are closer to the mean are more meaningful and hence we have a higher chance of rejecting a good sample.

4.4. Results after removing categories of samples from the training set on the gradient feature

Previously, we suspected that each category of samples removed from the training set has a unique effect on a classifier's recognition accuracy. To confirm this hypothesis, we have isolated each category to study their impact on the classifier. Each of the six categories: confusing pairs, legible, unrecognizable, poor, thick stroke, and very slanted have been individually removed from the training set. This process is performed on the training set using standard deviations ranging from 3.0 to 4.5. As a comparison, the unmodified training set using the same classifier yielded a recognition rate of 98.97%. The following table shows the results obtained from our experiment.

Table 9: Recognition accuracy and number of discarded samples by categories at different std dev from the training set (gradient feature).

Standard Deviation	4.5		4.0		3.5		3.0	
	Rejected samples	Recognition rate (%)	Rejected samples	Recognition rate (%)	Rejected samples	Recognition rate (%)	Rejected samples	Recognition rate (%)
Confusing pairs	1	98.97	3	98.97	9	98.98	22	98.99
Legible	27	98.97	37	98.96	69	98.95	165	98.93
Unrecognizable	5	98.97	7	98.98	13	98.99	23	98.96
Poor	4	98.99	7	98.98	13	98.98	37	98.99
Thick stroke	23	98.97	29	98.96	37	98.95	59	98.96
Very slanted	8	98.96	10	98.97	19	98.97	29	98.98

From the table above, we can observe that the confusing pairs and very slanted samples categories have a positive impact on the recognition rate as we reject more samples. As expected, discarding the legible samples has the opposite effect. The same trend can be observed on thick stroke samples but to a lesser extent. Removing illegible samples positively impacts the classifier's performance until we get to 3.5 standard deviations. A confusion matrix is shown below for the original accuracy of 98.97% obtained without any rejection.

Table 10: Confusion matrix for the original results without any rejection.

		Predicted Labels											
		0	1	2	3	4	5	6	7	8	9	Total	Acc. (%)
Actual Labels	0	977	0	0	0	0	0	0	0	3	0	980	99.69%
	1	0	1128	1	1	0	0	2	1	1	1	1135	99.38%
	2	0	0	1027	0	0	0	0	4	1	0	1032	99.52%
	3	0	0	4	998	0	3	0	0	4	1	1010	98.81%
	4	0	0	0	0	971	0	2	0	2	7	982	98.88%
	5	2	0	0	6	0	881	1	0	1	1	892	98.77%
	6	3	1	1	0	0	2	950	0	1	0	958	99.16%
	7	0	2	4	2	1	0	0	1014	1	4	1028	98.64%
	8	2	0	1	3	1	3	0	0	962	2	974	98.77%
	9	0	1	1	2	8	1	0	5	2	989	1009	98.02%
Total		984	1132	1039	1012	981	890	955	1024	978	1005		
Acc. (%)		99.29%	99.65%	98.85%	98.62%	98.98%	98.99%	99.48%	99.02%	98.36%	98.41%		

The confusion matrices for rejecting poor samples outside of 4.5 or 3.0 standard deviations and unrecognizable samples outside of 3.5 standard deviations are identical. The one for rejecting confusing pairs outside of 3.0 standard deviations differs slightly. These correspond to the best accuracy rates, which are all 98.99%. The following tables show these confusion matrices.

Table 11: Confusion matrix for rejecting poor samples outside 3.0 or 4.5 std dev. or unrecognizable samples outside of 3.5 std dev.

		Predicted Labels											
		0	1	2	3	4	5	6	7	8	9	Total	Acc. (%)
Actual Labels	0	977	0	0	0	0	0	0	0	3	0	980	99.69%
	1	0	1128	1	1	0	0	2	1	1	1	1135	99.38%
	2	0	0	1027	0	0	0	0	4	1	0	1032	99.52%
	3	0	0	4	999	0	3	0	0	3	1	1010	98.91%
	4	0	0	0	0	971	0	2	0	2	7	982	98.88%
	5	2	0	0	6	0	881	1	0	1	1	892	98.77%
	6	3	1	1	0	0	2	950	0	1	0	958	99.16%
	7	0	2	4	2	1	0	0	1014	1	4	1028	98.64%
	8	2	0	1	3	1	3	0	0	962	2	974	98.77%
	9	0	1	1	2	9	1	0	3	2	990	1009	98.12%
Total		984	1132	1039	1013	982	890	955	1022	977	1006		
Acc. (%)		99.29%	99.65%	98.85%	98.62%	98.88%	98.99%	99.48%	99.22%	98.46%	98.41%		

Table 12: Confusion matrix for rejecting confusion pairs outside 3.0 std dev.

		Predicted Labels											
		0	1	2	3	4	5	6	7	8	9	Total	Acc. (%)
Actual Labels	0	977	0	0	0	0	0	0	0	3	0	980	99.69%
	1	0	1128	1	1	0	0	2	1	1	1	1135	99.38%
	2	0	0	1027	0	0	0	0	4	1	0	1032	99.52%
	3	0	0	4	999	0	3	0	0	3	1	1010	98.91%
	4	0	0	0	0	971	0	2	0	2	7	982	98.88%
	5	2	0	0	6	0	882	1	0	1	0	892	98.88%
	6	3	1	1	0	0	2	950	0	1	0	958	99.16%
	7	0	2	4	2	1	0	0	1014	1	4	1028	98.64%
	8	2	0	1	3	1	3	0	0	962	2	974	98.77%
	9	0	1	1	2	9	1	0	4	2	989	1009	98.02%
Total		984	1132	1039	1013	982	891	955	1023	977	1004		
Acc. (%)		99.29%	99.65%	98.85%	98.62%	98.88%	98.99%	99.48%	99.12%	98.46%	98.51%		

Isolating the training set from poor samples consistently yield better results. Thus, it would be interesting to discard samples from multiple categories provided one of the categories is poor or confusing pairs. The following table shows the recognition rate of rejecting samples with certain combination of categories.

Table 13: Recognition accuracy and number of discarded samples from rejecting multiple categories in the training set (gradient feature) at different std dev.

Standard Deviation	4.5		4.0		3.5		3.0	
	Rejected samples	Recognition rate (%)	Rejected samples	Recognition rate (%)	Rejected samples	Recognition rate (%)	Rejected samples	Recognition rate (%)
confusing + poor	5	98.96	10	98.98	22	98.96	59	99.00
Leg + unrec + thick + slant	63	98.98	83	98.95	138	98.92	276	98.90
conf + unrec + poor	10	98.97	17	98.97	35	98.98	82	98.97
conf + unrec + thick	29	98.97	39	98.97	59	98.96	104	98.97
slant + poor + leg	39	98.98	54	98.97	101	98.94	231	98.95
unrec + poor	9	98.96	14	98.98	26	98.97	60	98.96

The above experiment yielded mixed results. The combination of confusing pairs, unrecognizable, and poor categories yielded promising results, as this set maintained a high recognition rate despite a large amount of rejected samples. Also, the highest accuracy was obtained using the confusing pairs and poor samples categories. The next table further expands on these results by adding another variable. We will use a different standard deviation for each individual category in order to select the outliers, which contribute to the rejection process. The confusion matrix for rejecting both poor samples and confusing pairs outside of 3.0 standard deviations is shown below.

Table 14: Confusion matrix of rejecting both poor samples and confusing pairs outside of 3.0 std dev.

		Predicted Labels											
		0	1	2	3	4	5	6	7	8	9	Total	Acc. (%)
Actual Labels	0	977	0	0	0	0	0	0	0	3	0	980	99.69%
	1	0	1128	1	1	0	0	2	1	1	1	1135	99.38%
	2	0	0	1027	0	0	0	0	4	1	0	1032	99.52%
	3	0	0	4	999	0	3	0	0	3	1	1010	98.91%
	4	0	0	0	0	972	0	2	0	2	6	982	98.98%
	5	2	0	0	6	0	881	1	0	1	1	892	98.77%
	6	3	1	1	0	0	2	950	0	1	0	958	99.16%
	7	0	2	4	2	1	0	0	1014	1	4	1028	98.64%
	8	2	0	1	3	1	3	0	0	962	2	974	98.77%
	9	0	1	1	2	9	1	0	3	2	990	1009	98.12%
Total		984	1132	1039	1013	983	890	955	1022	977	1005		
Acc. (%)		99.29%	99.65%	98.85%	98.62%	98.88%	98.99%	99.48%	99.22%	98.46%	98.51%		

Table 15: Results of the combination of confusing pairs, not legible, and poor sample categories where each individual category uses a different standard deviation for rejection.

Standard deviation			
confusing pairs	unrecognizable	poor	recognition rate (%)
3.00	3.50	4.50	99.00
3.50	3.50	4.50	98.97
3.50	4.00	4.50	98.99
3.50	4.50	4.50	98.97
3.50	3.50	4.00	98.97
3.50	3.50	3.50	98.98
3.50	3.50	3.00	98.98

The best choice would be to discard samples at 3.0 standard deviations for the confusing pairs category, 3.5 standard deviations for the unrecognizable category, and 4.5 standard deviations for the poor samples category. This correlates with the highest recognition rate achieved for each individual category at the standard deviations specified above. The confusion matrix is shown below.

Table 16: Confusion matrix of rejecting confusing pairs outside of 3.0 std dev., unrecognizable samples outside of 3.5 std dev., and poor samples outside of 4.5 std dev.

		Predicted Labels											
		0	1	2	3	4	5	6	7	8	9	Total	Acc. (%)
Actual Labels	0	977	0	0	0	0	0	0	0	3	0	980	99.69%
	1	0	1128	1	1	0	0	2	1	1	1	1135	99.38%
	2	0	0	1027	0	0	0	0	4	1	0	1032	99.52%
	3	0	0	4	1000	0	3	0	0	3	0	1010	99.01%
	4	0	0	0	0	971	0	2	0	2	7	982	98.88%
	5	2	0	0	6	0	882	1	0	1	0	892	98.88%
	6	3	1	1	0	0	2	950	0	1	0	958	99.16%
	7	0	2	4	2	1	0	0	1014	1	4	1028	98.64%
	8	2	0	1	4	1	3	0	0	961	2	974	98.67%
	9	0	1	1	2	9	1	0	3	2	990	1009	98.12%
Total		984	1132	1039	1015	982	891	955	1022	976	1004		
Acc. (%)		99.29%	99.65%	98.85%	98.52%	98.88%	98.99%	99.48%	99.22%	98.46%	98.61%		

4.5. Results after removing categories of samples from the training set on a combination of projection and gradient features

This experiment is repeated for a combination of both projection and gradient features. An accuracy rate of 99.01% is achieved using the new features set. The following table shows the results from our second experiment.

Table 17: Recognition accuracy and number of discarded samples by categories at different std dev. from the training set (gradient + projection features).

Standard Deviation	4.5		4.0		3.5		3.0	
	Rejected samples	Recognition rate (%)	Rejected samples	Recognition rate (%)	Rejected samples	Recognition rate (%)	Rejected samples	Recognition rate (%)
Confusing pairs	2	99.01	5	99.01	10	99.02	20	99.03
Legible	23	99.01	31	98.99	60	98.96	155	98.94
Unrecognizable	3	99.01	8	99.02	11	99.02	19	99.00
Poor	5	99.03	9	99.02	12	99.03	30	99.02
Thick stroke	26	99.00	31	99.01	36	98.99	51	98.97
Very slanted	5	99.02	12	99.00	18	99.01	22	99.02

We can observe a similar trend compared to our first experiment. The confusion matrices of the new results are shown below.

Table 18: Confusion matrix of results based on a combination of projection and gradient features.

		Predicted Labels											
		0	1	2	3	4	5	6	7	8	9	Total	Acc. (%)
Actual Labels	0	977	0	0	0	0	0	0	0	3	0	980	99.69%
	1	0	1126	1	1	0	0	2	1	3	1	1135	99.21%
	2	0	0	1028	0	0	0	0	4	0	0	1032	99.61%
	3	0	0	4	999	0	3	0	0	4	0	1010	98.91%
	4	0	0	0	0	971	0	2	0	2	7	982	98.88%
	5	2	0	0	6	0	881	1	0	1	1	892	98.77%
	6	2	1	1	0	0	3	949	0	2	0	958	99.06%
	7	0	2	4	2	1	0	0	1016	1	2	1028	98.83%
	8	2	0	1	3	1	3	0	0	962	2	974	98.77%
	9	0	0	1	2	8	1	0	4	1	992	1009	98.32%
Total		983	1129	1040	1013	981	891	954	1025	979	1005		
Acc. (%)		99.39%	99.73%	98.85%	98.62%	98.98%	98.88%	99.48%	99.12%	98.26%	98.71%		

Table 19: Confusion matrix of rejecting poor samples outside of 3.5 or 4.5 std dev. or rejecting confusing pairs outside of 3.0 std dev. on the combined training features set.

		Predicted Labels											
		0	1	2	3	4	5	6	7	8	9	Total	Acc. (%)
Actual Labels	0	977	0	0	0	0	0	0	0	3	0	980	99.69%
	1	0	1126	1	1	0	0	2	1	3	1	1135	99.21%
	2	0	0	1028	0	0	0	0	4	0	0	1032	99.61%
	3	0	0	4	999	0	3	0	0	4	0	1010	98.91%
	4	0	0	0	0	972	0	2	0	2	6	982	98.98%
	5	2	0	0	6	0	881	1	0	1	1	892	98.77%
	6	2	1	1	0	0	3	950	0	1	0	958	99.16%
	7	0	2	4	2	1	0	0	1016	1	2	1028	98.83%
	8	2	0	1	3	1	3	0	0	962	2	974	98.77%
	9	0	0	1	2	8	1	0	4	1	992	1009	98.32%
Total		983	1129	1040	1013	982	891	955	1025	978	1004		
Acc. (%)		99.39%	99.73%	98.85%	98.62%	98.98%	98.88%	99.48%	99.12%	98.36%	98.80%		

The confusing pairs, poor and unrecognizable categories have mostly similar or improved recognition rate.

Table 20: Recognition accuracy and number of discarded samples from rejecting multiple categories in the training set (gradient + projection features).

Standard Deviation	4.5		4.0		3.5		3.0	
	Rejected samples	Recognition rate (%)	Rejected samples	Recognition rate (%)	Rejected samples	Recognition rate (%)	Rejected samples	Recognition rate (%)
confusing + poor	7	99.00	14	99.02	22	99.04	50	99.03
Leg + unrec + thick + slant	57	99.01	82	98.98	125	98.95	247	98.90
conf + unrec + poor	10	99.01	22	99.02	33	99.02	69	99.02
conf + unrec + thick	31	99.00	44	99.01	57	98.99	90	99.00
slant + poor + leg	33	99.00	52	98.98	90	98.96	207	98.95
unrec + poor	8	99.01	17	99.02	23	99.01	49	98.98

Again, the combination of confusing pairs and poor samples yielded the highest improvement at a recognition rate of 99.04%. Below is the confusion matrix of rejecting confusing pairs and poor samples outside of 3.5 standard deviations.

Table 21: Confusion matrix of rejecting confusing pairs and poor samples outside of 3.5 std dev. on the combined training feature set.

		Predicted Labels											
		0	1	2	3	4	5	6	7	8	9	Total	Acc. (%)
Actual Labels	0	977	0	0	0	0	0	0	0	3	0	980	99.69%
	1	0	1128	1	1	0	0	2	1	1	1	1135	99.38%
	2	0	0	1028	0	0	0	0	4	0	0	1032	99.61%
	3	0	0	4	999	0	3	0	0	4	0	1010	98.91%
	4	0	0	0	0	971	0	2	0	2	7	982	98.88%
	5	2	0	0	6	0	881	1	0	1	1	892	98.77%
	6	2	1	1	0	0	2	950	0	2	0	958	99.16%
	7	0	2	4	2	1	0	0	1016	1	2	1028	98.83%
	8	2	0	1	3	1	3	0	0	962	2	974	98.77%
	9	0	0	1	2	8	1	0	4	1	992	1009	98.32%
Total		983	1131	1040	1013	981	890	955	1025	977	1005		
Acc. (%)		99.39%	99.73%	98.85%	98.62%	98.98%	98.99%	99.48%	99.12%	98.46%	98.71%		

Table 22: Results for the combination of confusing pairs, not legible, and poor sample categories where each individual category uses a different std dev. for rejection (gradient + projection features).

Standard deviation			
confusing pairs	unrecognizable	poor	recognition rate (%)
3.00	3.50	4.50	99.04
3.50	3.50	4.50	99.00
3.50	4.00	4.50	99.01
3.50	4.50	4.50	99.01
3.50	3.50	4.00	99.01
3.50	3.50	3.50	99.03
3.50	3.50	3.00	99.02

The best recognition rate is also achieved using a combination of the undesirable categories: confusing pairs, unrecognizable, and poor. Below is a confusion matrix of the combination yielding the best improvement in recognition accuracy.

Table 23: Confusion matrix of rejecting confusing pairs outside of 3.0 std dev, unrecognizable samples outside of 3.5 std dev. and poor samples outside of 4.5 std dev. on the combined training feature set.

		Predicted Labels											
		0	1	2	3	4	5	6	7	8	9	Total	Acc. (%)
Actual Labels	0	977	0	0	0	0	0	0	0	3	0	980	99.69%
	1	0	1128	1	1	0	0	2	1	1	1	1135	99.38%
	2	0	0	1028	0	0	0	0	4	0	0	1032	99.61%
	3	0	0	4	999	0	3	0	0	4	0	1010	98.91%
	4	0	0	0	0	972	0	2	0	2	6	982	98.98%
	5	2	0	0	6	0	881	1	0	1	1	892	98.77%
	6	2	1	1	0	0	3	949	0	2	0	958	99.06%
	7	0	2	4	2	1	0	0	1016	1	2	1028	98.83%
	8	2	0	1	3	1	3	0	0	962	2	974	98.77%
	9	0	0	1	2	8	1	0	4	1	992	1009	98.32%
Total		983	1131	1040	1013	982	891	954	1025	977	1004		
Acc. (%)		99.39%	99.73%	98.85%	98.62%	98.98%	98.88%	99.48%	99.12%	98.46%	98.80%		

These results correlate with those obtained from the previous experiment. We can conclude that better results can be obtained if we use the optimal rejection standard deviation for each category.

Chapter 5: Conclusion

This thesis' main focus was on recognizing offline unconstrained handwritten numerals. The motivation behind our research was to reduce errors from being made during the classification stage of a numeral recognition system. Errors are costly to correct. Thus, by gauging the expense (mainly accuracy rate) we should by all means eliminate any error made by the classifier. A two-stage rejection system was proposed to mainly improve the reliability aspect of a classifier. Our experiment studied its effect on both recognition rate and substitution rate by employing different types of features such as statistical and structural. The study was conducted over the popular MNIST database for easier comparison with other methods. We used a support vector machines based classifier, as it still provides the core to one of the most performing recognition systems to date.

5.1. Findings

My thesis focuses on a novel rejection approach that increases the reliability of the current classification systems. By improving the classification system's error prevention and tolerance capabilities, we can build a dependable classifier and minimize its substitution rate. The main findings can be summarized as follows:

5.1.1. The effect of discarding samples in the training set only for a statistical feature

If too many training samples were removed, the accuracy was expected to suffer. The classifier will not have sufficient quality training samples and testing samples to recognize samples correctly using the limited data. It was noticed that discarding excessively more samples does impair the classifier's effectiveness to identify samples. Using a rejection criterion less than three standard deviations greatly diminishes the recognition rate of the classifier.

The accuracy was observed to be very close to the original accuracy when training samples outside of three standard deviations or more are removed. For the tested standard deviation range of 3.0 to 4.5, 19 out of 25 standard deviation values used yielded equal or better accuracy.

In most cases, rejecting outliers in the training set proved to increase accuracy results.

There is no trivial way to find the optimal standard deviation to use on different feature sets and variability of the training set. However, it is useful to eliminate mislabeled samples by making the overall classification system more robust. The rejection step preceding training ensures that only valid samples are handed down to the classifier. The system can be error tolerant and shielded from errors, which are initially introduced.

5.1.2. The effect of discarding samples in the training and testing sets.

If too many samples are removed from the training set, the classifier's accuracy will drop. This effect is obvious when samples are rejected from the testing set. It is unavoidable that some of the correctly recognized samples will be discarded along with the outliers. Thus, it is important that we reject only the real outliers from both the training and testing sets. Choosing a relatively large value for the rejection criterion or the standard deviation can limit the number of good samples being rejected, but at the expense of a higher substitution rate.

As observed in the current thesis, rejecting in the standard deviation range of 3.0 to 4.5 standard deviations for the training set yielded the best recognition accuracy rate. Similarly, improvement in the substitution rate is noticeable for rejection in the standard deviation range of 3.0 to 4.5 in the testing set. In most cases, rejecting outliers in the training and testing sets can provide lower substitution rates.

No increase in accuracy is possible by eliminating samples from the testing set. The accuracy can only decrease, because accurately predicted samples will eventually get rejected with the outliers. Nevertheless, by excluding the outliers outside four standard deviations, we obtained accuracies that are very close to the original and slightly lower substitution rates. Furthermore, a more sophisticated approach for the testing set would yield superior results in terms of recognition and substitution rates. The system can be made error tolerant by discarding samples in the training set and the error prevention mechanism works by rejecting outlying predicted samples from the testing set.

5.1.3. The effect of discarding samples in the training set with the gradient feature

The results are close to the original for a rejection range greater than 3.9 standard deviations with no observable improvement. In contrast to the projection feature tested, there were improvements when the standard deviation used was greater than 3.7 standard deviations.

Many reasons may explain the results of the tests with the gradient feature. First, this feature is very accurate and information rich. By removing samples from this feature set containing only relevant information would impair the classifier's recognition rate.

Useful samples being discarded with the bad samples can be another cause. As a result, it nullifies any improvement that could have been detected. To verify this hypothesis, an examination of the samples rejected was done.

5.1.4. Categorizing the rejected samples from the training set.

We scrutinized the samples that are excluded from the training set when different standard deviations are used as rejection criteria. These rejected training samples have been classified into different categories. The samples have been categorized into six major groups: good samples, very slanted samples, thick stroke samples, poor samples, unrecognizable samples, and confusing pairs samples. The first three categories defined are desirable samples, and the last three are defined as undesirable samples from the training set.

After grouping the samples into their respective categories, our original hypothesis was supported. The number of desirable samples is almost always greater than the number of undesirable samples in the rejected samples set. This percentage grows as more samples are rejected using a smaller standard deviation value for our rejection criterion. The classifier's performance deteriorates, as the percentage of desirable samples is more than double of the undesirable samples in the whole rejected samples set. Samples closer to the mean are more likely to be accurate and the probability of removing a good sample is greater.

5.1.5. The effect of discarding samples in the training set with the combination of gradient and projection features

The results obtained are very similar to those from the gradient feature alone. We have studied the effect of discarding categories of samples from the gradient feature and a combination of gradient and projection features. The results revealed that blindly rejecting samples from the training set using the same standard deviation criterion for each category did not yield any improvement. However, by adjusting the standard deviation criteria for each category of samples rejected to an optimal value, we were able to achieve improvements of up to 0.04% in the accuracy rate.

5.2. Future Work

There is not a perfect solution for all the problems encountered in pattern recognition. The solution we proposed improves a classifier's reliability by implementing an error tolerant and preventative mechanism.

The error prevention part of our system can be further modified. An entirely new approach can be taken to perform the traditional post recognition rejection. Ideally, it will reduce the number of valid predictions rejected in the process. Moreover, a method for optimizing the removal of outliers in the training set can be developed to achieve the better results that we expect.

Our solution can also be tested against different databases and field such as character recognition. The error tolerant method can be easily added as a module to almost any existing classification system.

Bibliography

[1] J. Duan, M. Y. Zhang, and Q. Li, "A multi-stage adaptive binaraziation scheme for document images," *Proc. International Joint Conference on Computational Sciences and Optimization*, Hainan, pp. 867-896, 2009.

[2] F. De Zeeuw, "Slant correction using histograms", Bachelor thesis in artificial intelligence, Rijks University, 2006.

[3] C. L. He, J. X. Dong, P. Zhang, and C. Y. Suen, "The role of size normalization on the recognition rate of handwritten numerals," *Proc. IAPR TC3 Workshop of 8th Internationl Confererence on Document Analysis and Recognition*, Seoul, pp. 8-12,2005.

[4] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Communcations of the ACM*, vol. 27, no. 3, pp. 236-239, 1984.

[5] S. Kambar, "Generating synthetic data by morphing transformation for handwritten numeral recogntion with v-SVM", Master Thesis, Concordia University, 2005.

[6] C. L. Liu, H. Sako, and H. Fujisawa, "Performance evaluation of pattern classifiers for handwritten character recognition," *International Journal on Document Analysis and Recognition*, vol. 4, pp. 191-204, 2002.

[7] D. C. Ciresan, U. Meier, and J. Scmidhuber, "Multi-column deep neural networks for image classification," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Providence, pp.1-20, 2012.

[8] J.X. Dong, A. Krzyzak, and C. Y. Suen, "A fast SVM training algorithm," in *International Workshop on Pattern Recognition with Support Vector Machines*, Niagara Falls, pp. 53-67, 2002.

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol.86 pp. 2278-2324, 1998.

[10] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," *Proc. 7th International Conference on Document Analysis and Recognition*, Washington, pp. 958-963, 2003.

[11] C. L. He and C. Y. Suen, "A hybrid multiple classifier system of unconstrained handwritten numeral recognition," *Pattern Recognition and Image Analysis*, vol. 17, no. 4, pp. 608-611, 2007.

[12] C. Y. Suen and J. Tan, "Analysis of errors of handwritten digits made by a multitude of classifiers," *Pattern Recognition Letters*, vol. 26, no. 3, pp. 369-379, 2005.

[13] Y. Lecun and C. Cortes. The MNIST Database of Handwritten Digits. [Online]. "http://yann.lecun.com/exdb/mnist/".

[14] N. Otsu, "A threshold selection method from gray-level histograms," in *IEEE Trans. Systems, Man and Cybernatics*, pp. 62–66, 1979.

[15] C. L. He, Reliable pattern recognition system with novel semi-supervised learning approach, Doctoral Thesis, Concordia University, 2010.

[16] M. Shi, Y. Fujisawa, T. Wakabayashi, and F. Kimura, "Handwritten numeral recognition using gradient and curvature of gray scale image," *Pattern Recognition*, vol. 35, no. 10, pp. 2051-2059, 2002.

[17] Cyc. (2008, February) Wikipedia. [Online]. "http://en.wikipedia.org/wiki/Support_vector_machine".

[18] C. C. Chang and C. J. Lin. (2001) LIBSVM: a library for support vector machines. [Online]. "http://www.csie.ntu.edu.tw/~cjlin/libsvm".

[19] C. L. He, P. Zhang, J. X. Dong, C. Y. Suen, and T. D. Bui, "The Role of Size Normalization on the Recognition Rate of Handwritten Numerals," in *Proceedings of IAPR TC3 Workshop of 8th International Conference on Document Analysis and Recognition: Neural Networks and Learning in Document Analysis and Recognition*, Seoul, pp. 8-12, 2005.

[20] X. X. Niu, and C. Y. Suen " A novel hybrid CNN-SVM classifier for recognizing handwritten digits" *Pattern Recognition,* Seoul, vol. 45, no.1, pp. 1318-1325, 2012.

[21] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Third Edition ed., Pearson Education, Ed. Upper Saddle River, USA: Pearson Prentice Hall, 2008.

APPENDIX: Results of using different standard deviations for rejection in both training and testing sets on the projection feature.

training standard deviation	testing standard deviation	accuracy rate (%)	rejected samples in training	rejected samples in testing	substitution rate (%)
2.0	1.0	93.9019	2836	2637	6.0981
2.0	1.5	93.2491	2836	979	6.7509
2.0	2.0	92.8681	2836	255	7.1319
2.0	2.5	92.7346	2836	35	7.2654
2.0	3.0	92.7456	2836	6	7.2544
2.0	3.5	92.7385	2836	2	7.2615
2.0	4.0	92.7393	2836	1	7.2607
2.0	4.5	92.7400	2836	0	7.2600
2.1	1.0	93.9208	2190	2647	6.0792
2.1	1.5	93.2823	2190	994	6.7177
2.1	2.0	92.9107	2190	267	7.0893
2.1	2.5	92.7223	2190	38	7.2777
2.1	3.0	92.7156	2190	6	7.2844
2.1	3.5	92.7085	2190	2	7.2915
2.1	4.0	92.7093	2190	1	7.2907
2.1	4.5	92.7100	2190	0	7.2900
2.2	1.0	94.0272	1770	2650	5.9728
2.2	1.5	93.3896	1770	999	6.6104
2.2	2.0	93.0776	1770	278	6.9224
2.2	2.5	92.8780	1770	45	7.1220
2.2	3.0	92.8636	1770	9	7.1364

2.2	3.5	92.8579	1770	3	7.1421
2.2	4.0	92.8500	1770	0	7.1500
2.2	4.5	92.8500	1770	0	7.1500
2.3	1.0	94.1529	1467	2663	5.8471
2.3	1.5	93.5247	1467	1012	6.4753
2.3	2.0	93.1823	1467	290	6.8177
2.3	2.5	92.9576	1467	46	7.0424
2.3	3.0	92.9537	1467	9	7.0463
2.3	3.5	92.9472	1467	4	7.0528
2.3	4.0	92.9393	1467	1	7.0607
2.3	4.5	92.9400	1467	0	7.0600
2.4	1.0	94.2557	1213	2671	5.7443
2.4	1.5	93.6080	1213	1020	6.3920
2.4	2.0	93.2310	1213	294	6.7690
2.4	2.5	92.9964	1213	48	7.0036
2.4	3.0	92.9930	1213	10	7.0070
2.4	3.5	92.9772	1213	4	7.0228
2.4	4.0	92.9693	1213	1	7.0307
2.4	4.5	92.9700	1213	0	7.0300
2.5	1.0	94.3345	1004	2675	5.6655
2.5	1.5	93.6817	1004	1026	6.3183
2.5	2.0	93.3375	1004	304	6.6625
2.5	2.5	93.1093	1004	59	6.8907
2.5	3.0	93.1117	1004	12	6.8883
2.5	3.5	93.0972	1004	4	6.9028
2.5	4.0	93.0893	1004	1	6.9107
2.5	4.5	93.0900	1004	0	6.9100

2.6	1.0	94.3958	811	2684	5.6042
2.6	1.5	93.7214	811	1033	6.2786
2.6	2.0	93.3574	811	305	6.6426
2.6	2.5	93.1482	811	61	6.8518
2.6	3.0	93.1511	811	13	6.8489
2.6	3.5	93.1366	811	5	6.8634
2.6	4.0	93.1286	811	2	6.8714
2.6	4.5	93.1293	811	1	6.8707
2.7	1.0	94.4726	690	2691	5.5274
2.7	1.5	93.7960	690	1038	6.2040
2.7	2.0	93.4276	690	308	6.5724
2.7	2.5	93.2166	690	64	6.7834
2.7	3.0	93.2212	690	13	6.7788
2.7	3.5	93.2066	690	5	6.7934
2.7	4.0	93.1986	690	2	6.8014
2.7	4.5	93.1993	690	1	6.8007
2.8	1.0	94.4596	522	2690	5.5404
2.8	1.5	93.7967	522	1037	6.2033
2.8	2.0	93.4469	522	310	6.5531
2.8	2.5	93.2454	522	66	6.7546
2.8	3.0	93.2305	522	14	6.7695
2.8	3.5	93.2066	522	5	6.7934
2.8	4.0	93.1986	522	2	6.8014
2.8	4.5	93.1993	522	1	6.8007
2.9	1.0	94.4855	429	2692	5.5145
2.9	1.5	93.8288	429	1039	6.1712
2.9	2.0	93.4688	429	308	6.5312

2.9	2.5	93.2857	429	66	6.7143
2.9	3.0	93.2799	429	15	6.7201
2.9	3.5	93.2560	429	6	6.7440
2.9	4.0	93.2480	429	3	6.7520
2.9	4.5	93.2493	429	1	6.7507
3.0	1.0	94.5129	379	2692	5.4871
3.0	1.5	93.8400	379	1039	6.1600
3.0	2.0	93.4888	379	309	6.5112
3.0	2.5	93.2958	379	66	6.7042
3.0	3.0	93.2806	379	14	6.7194
3.0	3.5	93.2566	379	5	6.7434
3.0	4.0	93.2486	379	2	6.7514
3.0	4.5	93.2493	379	1	6.7507
3.1	1.0	94.4718	305	2692	5.5282
3.1	1.5	93.8281	305	1040	6.1719
3.1	2.0	93.4991	305	309	6.5009
3.1	2.5	93.2964	305	65	6.7036
3.1	3.0	93.2906	305	14	6.7094
3.1	3.5	93.2666	305	5	6.7334
3.1	4.0	93.2587	305	2	6.7413
3.1	4.5	93.2593	305	1	6.7407
3.2	1.0	94.4855	274	2692	5.5145
3.2	1.5	93.8393	274	1040	6.1607
3.2	2.0	93.5198	274	309	6.4802
3.2	2.5	93.3166	274	65	6.6834
3.2	3.0	93.3100	274	15	6.6900
3.2	3.5	93.2860	274	6	6.7140

3.2	4.0	93.2787	274	2	6.7213
3.2	4.5	93.2793	274	1	6.7207
3.3	1.0	94.4718	237	2692	5.5282
3.3	1.5	93.8491	237	1042	6.1509
3.3	2.0	93.5294	237	310	6.4706
3.3	2.5	93.3260	237	66	6.6740
3.3	3.0	93.3193	237	16	6.6807
3.3	3.5	93.2953	237	7	6.7047
3.3	4.0	93.2880	237	3	6.7120
3.3	4.5	93.2893	237	1	6.7107
3.4	1.0	94.4581	205	2692	5.5419
3.4	1.5	93.8372	205	1043	6.1628
3.4	2.0	93.5384	205	312	6.4616
3.4	2.5	93.3347	205	68	6.6653
3.4	3.0	93.3193	205	16	6.6807
3.4	3.5	93.2953	205	7	6.7047
3.4	4.0	93.2880	205	3	6.7120
3.4	4.5	93.2893	205	1	6.7107
3.5	1.0	94.4422	177	2695	5.5578
3.5	1.5	93.8135	177	1045	6.1865
3.5	2.0	93.5157	177	315	6.4843
3.5	2.5	93.3038	177	69	6.6962
3.5	3.0	93.2792	177	16	6.7208
3.5	3.5	93.2553	177	7	6.7447
3.5	4.0	93.2480	177	3	6.7520
3.5	4.5	93.2493	177	1	6.7507
3.6	1.0	94.4308	160	2692	5.5692

3.6	1.5	93.8135	160	1045	6.1865
3.6	2.0	93.5157	160	315	6.4843
3.6	2.5	93.3132	160	70	6.6868
3.6	3.0	93.2893	160	16	6.7107
3.6	3.5	93.2653	160	7	6.7347
3.6	4.0	93.2580	160	3	6.7420
3.6	4.5	93.2593	160	1	6.7407
3.7	1.0	94.4308	148	2692	5.5692
3.7	1.5	93.8240	148	1046	6.1760
3.7	2.0	93.5364	148	315	6.4636
3.7	2.5	93.3333	148	70	6.6667
3.7	3.0	93.3093	148	16	6.6907
3.7	3.5	93.2853	148	7	6.7147
3.7	4.0	93.2780	148	3	6.7220
3.7	4.5	93.2793	148	1	6.7207
3.8	1.0	94.4437	119	2693	5.5563
3.8	1.5	93.8470	119	1045	6.1530
3.8	2.0	93.5854	119	319	6.4146
3.8	2.5	93.3723	119	72	6.6277
3.8	3.0	93.3474	119	19	6.6526
3.8	3.5	93.3147	119	8	6.6853
3.8	4.0	93.3073	119	4	6.6927
3.8	4.5	93.2993	119	1	6.7007
3.9	1.0	94.4323	106	2690	5.5677
3.9	1.5	93.8372	106	1043	6.1628
3.9	2.0	93.5564	106	316	6.4436
3.9	2.5	93.3723	106	72	6.6277

3.9	3.0	93.3474	106	19	6.6526
3.9	3.5	93.3147	106	8	6.6853
3.9	4.0	93.3073	106	4	6.6927
3.9	4.5	93.2993	106	1	6.7007
4.0	1.0	94.3905	99	2691	5.6095
4.0	1.5	93.8023	99	1045	6.1977
4.0	2.0	93.5247	99	317	6.4753
4.0	2.5	93.3414	99	73	6.6586
4.0	3.0	93.3166	99	20	6.6834
4.0	3.5	93.2846	99	8	6.7154
4.0	4.0	93.2773	99	4	6.7227
4.0	4.5	93.2693	99	1	6.7307
4.1	1.0	94.3897	83	2692	5.6103
4.1	1.5	93.7814	83	1043	6.2186
4.1	2.0	93.5344	83	318	6.4656
4.1	2.5	93.3414	83	73	6.6586
4.1	3.0	93.3166	83	20	6.6834
4.1	3.5	93.2840	83	9	6.7160
4.1	4.0	93.2766	83	5	6.7234
4.1	4.5	93.2693	83	1	6.7307
4.2	1.0	94.3631	79	2691	5.6369
4.2	1.5	93.7702	79	1043	6.2298
4.2	2.0	93.5027	79	319	6.4973
4.2	2.5	93.3199	79	75	6.6801
4.2	3.0	93.2866	79	20	6.7134
4.2	3.5	93.2539	79	9	6.7461
4.2	4.0	93.2466	79	5	6.7534

4.2	4.5	93.2393	79	1	6.7607
4.3	1.0	94.3768	77	2691	5.6232
4.3	1.5	93.7814	77	1043	6.2186
4.3	2.0	93.5137	77	318	6.4863
4.3	2.5	93.3306	77	74	6.6694
4.3	3.0	93.2973	77	19	6.7027
4.3	3.5	93.2646	77	8	6.7354
4.3	4.0	93.2566	77	5	6.7434
4.3	4.5	93.2493	77	1	6.7507
4.4	1.0	94.3905	67	2691	5.6095
4.4	1.5	93.8030	67	1044	6.1970
4.4	2.0	93.5441	67	319	6.4559
4.4	2.5	93.3508	67	74	6.6492
4.4	3.0	93.3260	67	21	6.6740
4.4	3.5	93.2933	67	10	6.7067
4.4	4.0	93.2860	67	6	6.7140
4.4	4.5	93.2787	67	2	6.7213
4.5	1.0	94.4163	65	2693	5.5837
4.5	1.5	93.8135	65	1045	6.1865
4.5	2.0	93.5537	65	320	6.4463
4.5	2.5	93.3602	65	75	6.6398
4.5	3.0	93.3360	65	21	6.6640
4.5	3.5	93.3033	65	10	6.6967
4.5	4.0	93.2960	65	6	6.7040
4.5	4.5	93.2887	65	2	6.7113